HP ProCurve Switch Software

Advanced Traffic Management Guide

3500 switches
3500yl switches
5400zl switches
6200yl switches
6600 switches
8200zl switches

Software version K.14.52
March 2010

HP ProCurve

3500 Switches
3500yl Switches
5400zl Switches
6200yl Switch
6600 Switches
8200zl Switches

March 2010
K.14.52

Advanced Traffic Management Guide

© Copyright 2005–2010 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. All Rights Reserved.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard.

Publication Number

5992-3060
March 2010

Applicable Products

HP ProCurve Switch 3500-24	(J9470A)
HP ProCurve Switch 3500-48	(J9472A)
HP ProCurve Switch 3500-24 PoE	(J9471A)
HP ProCurve Switch 3500-48 PoE	(J9473A)
HP ProCurve Switch 3500yl-24G-PWR	(J8692A)
HP ProCurve Switch 3500yl-48G-PWR	(J8693A)
HP ProCurve Switch 3500yl-24G-PoE+	(J9310A)
HP ProCurve Switch 3500yl-48G-PoE+	(J9311A)
HP ProCurve Switch 5406zl	(J8697A)
HP ProCurve Switch 5412zl	(J8698A)
HP ProCurve Switch 6200yl-24G	(J8992A)
HP ProCurve Switch 8206zl	(J9475A)
HP ProCurve Switch 8212zl	(J8715A/B)
HP ProCurve Switch 6600-24G	(J9263A)
HP ProCurve Switch 6600-24G-4XG	(J9264A)
HP ProCurve Switch 6600-24G-24XG	(J9265A)
HP ProCurve Switch 6600-48G	(J9451A)
HP ProCurve Switch 6600-48G-4XG	(J9452A)
HP ProCurve 24-Port 10/100/1000 PoE+ zl Module	(J9307A)
HP ProCurve 20-Port 10/100/1000 PoE+/4-Port MiniGBIC zl Module	(J9308A)
HP ProCurve 4-Port 10GbE SFP+ zl Module	(J9309A)
HP ProCurve 24-Port 10/100 PoE+ zl Module	(J9478A)

Trademark Credits

Microsoft, Windows, and Microsoft Windows NT are US registered trademarks of Microsoft Corporation.

Disclaimer

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD COMPANY MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Warranty

See the Customer Support/Warranty booklet included with the product.

A copy of the specific warranty terms applicable to your Hewlett-Packard products and replacement parts can be obtained from your HP Sales and Service Office or authorized dealer.

Hewlett-Packard Company
8000 Foothills Boulevard, m/s 5551
Roseville, California 95747-5551
www.procurve.com

Contents

Product Documentation

About Your Switch Manual Set xv
- Printed Publications. xv
- Electronic Publications xv

Software Feature Index xvi

1 Getting Started

Contents 1-1

Introduction 1-2

Conventions 1-2
- Command Syntax Statements 1-2
- Command Prompts 1-3
- Screen Simulations 1-3
- Configuration and Operation Examples 1-3
- Keys 1-4

Sources for More Information 1-4
- Getting Documentation From the Web 1-6
- Online Help 1-6
  - Menu Interface 1-6
  - Command Line Interface 1-7
  - WebAgent (Web Browser Interface) 1-7

Need Only a Quick Start? 1-7
- IP Addressing 1-7

To Set Up and Install the Switch in Your Network 1-8
- Physical Installation 1-8

2 Static Virtual LANs (VLANs)

Contents 2-1

Overview 2-4

Introduction 2-5
General VLAN Operation 2-5
Types of Static VLANs Available in the Switch 2-6
Port-Based VLANs 2-6
Protocol-Based VLANs 2-6
Designated VLANs 2-6
Terminology 2-7
Static VLAN Operation 2-8
VLAN Environments 2-9
VLAN Operation 2-10
Routing Options for VLANs 2-11
Overlapping (Tagged) VLANs 2-12
Per-Port Static VLAN Configuration Options 2-14
VLAN Operating Rules 2-15
General Steps for Using VLANs 2-19
Multiple VLAN Considerations 2-20
Single Forwarding Database Operation 2-21
Example of an Unsupported Configuration and How To Correct It 2-22
Multiple Forwarding Database Operation 2-23
Configuring VLANs 2-24
Menu: Configuring Port-Based VLAN Parameters 2-24
To Change VLAN Support Settings 2-25
Adding or Editing VLAN Names 2-26
Adding or Changing a VLAN Port Assignment 2-28
CLI: Configuring Port-Based and Protocol-Based VLAN Parameters 2-30
Customizing the Show VLANs Output 2-36
Creating an Alias for Show VLAN Commands 2-38
Note on Using Pattern Matching with the "Show VLANs Custom" Command 2-39
Changing the Number of VLANs Allowed on the Switch 2-39
Web: Viewing and Configuring VLAN Parameters 2-45
802.1Q VLAN Tagging 2-46
Special VLAN Types 2-51
VLAN Support and the Default VLAN 2-51

The Primary VLAN 2-51
The Secure Management VLAN 2-52
Preparation 2-54
Configuration 2-55
Using DHCP to Obtain an IP Address 2-56
Deleting the Management VLAN 2-59
Operating Notes for Management VLANs 2-59
Voice VLANs 2-60
Operating Rules for Voice VLANs 2-60
Components of Voice VLAN Operation 2-61
Voice VLAN QoS Prioritizing (Optional) 2-61
Voice VLAN Access Security 2-62
Effect of VLANs on Other Switch Features 2-62
Spanning Tree Operation with VLANs 2-62
IP Interfaces 2-63
VLAN MAC Address 2-63
Port Trunks 2-63
Port Monitoring 2-63
Jumbo Packet Support 2-63
VLAN Restrictions 2-64
Migrating Layer 3 VLANs Using VLAN MAC Configuration 2-65
VLAN MAC Address Reconfiguration 2-65
Handling Incoming and Outgoing VLAN Traffic 2-66
Sending Heartbeat Packets with a Configured MAC Address 2-67
Configuring a VLAN MAC Address with Heartbeat Interval 2-68
Operating Notes 2-68
Example 2-69
Verifying a VLAN MAC Address Configuration 2-69

3 GVRP

Contents 3-1
Overview 3-2
Introduction 3-3
General Operation 3-4

Per-Port Options for Handling GVRP "Unknown VLANs" 3-7

Per-Port Options for Dynamic VLAN Advertising and Joining 3-9

GVRP and VLAN Access Control 3-11

Advertisements and Dynamic Joins 3-11

Port-Leave From a Dynamic VLAN 3-11

Planning for GVRP Operation 3-12

Configuring GVRP On a Switch 3-13

Menu: Viewing and Configuring GVRP 3-13

CLI: Viewing and Configuring GVRP 3-14

Web: Viewing and Configuring GVRP 3-18

GVRP Operating Notes 3-18

4 Multiple Instance Spanning-Tree Operation

Contents 4-1

Overview 4-3

802.1s Multiple Spanning Tree Protocol (MSTP) 4-6

MSTP Structure 4-7

How MSTP Operates 4-9

MST Regions 4-9

Regions, Legacy STP and RSTP Switches, and the Common Spanning Tree (CST) 4-11

MSTP Operation with 802.1Q VLANs 4-12

Terminology 4-13

Operating Rules 4-15

MSTP Compatibility with RSTP or STP 4-16

Configuring MSTP 4-17

Planning an MSTP Application 4-17

MSTP Configuration Overview 4-19

Configuring MSTP Operation Mode and Global Settings 4-21

Configuring MSTP Per-Port Parameters 4-26

Configuring Per Port Parameters 4-27

Configuring BPDU Filtering 4-30

Configuring BPDU Protection 4-31

PVST Protection and Filtering 4-34

Configuring MST Instance Parameters 4-39
Configuring MST Instance Per-Port Parameters 4-41
Enabling or Disabling Spanning Tree Operation 4-44
Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another 4-44
MSTP VLAN Configuration Enhancement 4-46
PreConfiguring VLANs in an MST Instance 4-47
Configuring MSTP Instances with the VLAN Range Option 4-48
Operating Notes for the VLAN Configuration Enhancement . . . 4-50
How to Save Your Current Configuration 4-51
Displaying MSTP Statistics and Configuration 4-53
Displaying Global MSTP Status 4-54
Displaying Detailed Port Information 4-56
Displaying Status for a Specific MST Instance 4-57
Displaying the MSTP Configuration 4-58
Troubleshooting an MSTP Configuration 4-62
Displaying the Change History of Root Bridges 4-62
Displaying Debug Counters for All MST Instances 4-65
Displaying Debug Counters for One MST Instance 4-66
Displaying Debug Counters for Ports in an MST Instance 4-68
Field Descriptions in MSTP Debug Command Output 4-70
Troubleshooting MSTP Operation 4-73
Loop Protection 4-74
Configuring Loop Protection 4-75
Viewing Loop Protection Status 4-76

5 Switch Meshing

Contents 5-1
Introduction 5-2
Switch Meshing Fundamentals 5-4
Terminology 5-4
Operating Rules 5-5
Using a Heterogeneous Switch Mesh 5-7
Bringing Up a Switch Mesh Domain 5-8

Further Operating Information 5-8
Configuring Switch Meshing 5-9
Preparation 5-9
Menu: To Configure Switch Meshing 5-9
CLI: To View and Configure Switch Meshing 5-12
Viewing Switch Mesh Status 5-12
CLI: Configuring Switch Meshing 5-14
Operating Notes for Switch Meshing 5-15
Flooded Traffic 5-16
Unicast Packets with Unknown Destinations 5-17
Spanning Tree Operation with Switch Meshing 5-17
Filtering/Security in Meshed Switches 5-20
IP Multicast (IGMP) in Meshed Switches 5-20
Static VLANs 5-20
Dynamic VLANs 5-21
Jumbo Packets 5-21
Mesh Design Optimization 5-22
Other Requirements and Restrictions 5-23

6 Quality of Service: Managing Bandwidth More Effectively

Contents 6-1
Using Quality of Service Policies 6-4
QoS Terminology 6-7
QoS Operation 6-9
Globally-Configured QoS 6-10
Classifier-Based QoS 6-11
QoS Packet Classification 6-12
Globally-Configured Packet Classification 6-12
Classifier-Based Match Criteria 6-13
QoS Traffic Marking 6-14
Globally-Configured Traffic Marking 6-14
Layer 2 802.1p Prioritization 6-14
Layer 3 DSCP Marking 6-16
VLAN and Untagged VLAN Environments 6-17

Classifier-Based Traffic Marking 6-18
Globally-Configured QoS 6-19
Global QoS Configuration Procedure 6-19
Viewing a Global QoS Configuration 6-21
Global QoS Restrictions 6-22
Global TCP/UDP Classifier 6-24
Assigning an 802.1p Priority for a Global TCP/UDP Classifier . 6-25
Operating Notes on Using TCP/UDP Port Ranges 6-26
Assigning a DSCP Policy for a Global TCP/UDP Classifier 6-27
Displaying Resource Usage for QoS Policies 6-32
Global IP-Device Classifier 6-33
Assigning a Priority for a Global IP-Device Classifier 6-34
Assigning a DSCP Policy For a Global IP-Device Classifier 6-36
Global IP Type-of-Service Classifier 6-41
IPv4 ToS/IPv6 Traffic Class Byte 6-42
Assigning an 802.1p Priority for a Global IP-Precedence Classifier 6-44
Assigning an 802.1p Priority for a Global IP-Diffserv Classifier 6-45
Assigning a DSCP Policy for a Global IP-Diffserv Classifier . . . 6-49
Comparison of Global IP Type-of-Service Classifiers 6-53
Global Layer-3 Protocol Classifier 6-54
Assigning a Priority for a Global Layer-3 Protocol Classifier . . . 6-54
Global VLAN-ID Classifier 6-56
Assigning a Priority for a Global VLAN-ID Classifier 6-56
Assigning a DSCP Policy for a Global VLAN-ID Classifier 6-58
Global Source-Port Classifier 6-62
Assigning a Priority for a Global Source-Port Classifier 6-62
Assigning a DSCP Policy for a Global Source-Port Classifier . . 6-64
IP Multicast (IGMP) Interaction with QoS 6-69
Advanced Classifier-Based QoS 6-70
Classifier-Based QoS Model 6-71
Classifier-Based QoS Configuration Procedure 6-71
Configuring QoS Actions in a Policy 6-76
Override of Global QoS Settings 6-81
Viewing a Classifier-Based QoS Configuration 6-81

Classifier-Based QoS Restrictions 6-85
Interaction with Other Software Features 6-85
Classifier-Based QoS Configuration Examples 6-86
QoS Policy for Layer 4 TCP/UDP Traffic 6-87
QoS Policy for Subnet Traffic 6-87
Differentiated Services Codepoint (DSCP) Mapping 6-88
Default Priority Settings for Selected Codepoints 6-89
Displaying Non-Default Codepoint Settings 6-90
Notes on Changing a Priority Setting 6-91
Error Messages for DSCP Policy Changes 6-92
Example of Changing the Priority Setting on a Policy When One or More Classifiers Are Currently Using the Policy . 6-92
QoS Queue Configuration 6-95
Configuring the Number of Priority Queues 6-96
Viewing the QoS Queue Configuration 6-98

7 Stack Management for the 3500, 3500yl, 6200yl and 6600 Switches

Contents 7-1
Introduction to Stack Management on the 3500, 3500yl, 6200yl and 6600 Switches 7-3
Components of ProCurve Stack Management 7-5
General Stacking Operation 7-5
Operating Rules for Stacking 7-7
General Rules 7-7
Specific Rules 7-8
Configuring Stack Management 7-9
Overview of Configuring and Bringing Up a Stack 7-9
General Steps for Creating a Stack 7-11
Using the Menu Interface To View Stack Status and Configure Stacking 7-13
Using the Menu Interface To View and Configure a Commander Switch 7-13
Using the Menu To Manage a Candidate Switch 7-15
Using the Commander To Manage The Stack 7-17

Using the Commander To Access Member Switches for Configuration Changes and Monitoring Traffic 7-23
Converting a Commander or Member to a Member of Another Stack 7-24
Monitoring Stack Status 7-25
Using the CLI To View Stack Status and Configure Stacking 7-29
Using the CLI To View Stack Status 7-31
Using the CLI To Configure a Commander Switch 7-33
Adding to a Stack or Moving Switches Between Stacks 7-35
Using the CLI To Remove a Member from a Stack 7-40
Using the CLI To Access Member Switches for Configuration Changes and Traffic Monitoring 7-42
SNMP Community Operation in a Stack 7-43
Using the CLI To Disable or Re-Enable Stacking 7-44
Transmission Interval 7-44
Stacking Operation with Multiple VLANs Configured 7-44
Status Messages 7-45

8 QinQ (Provider Bridging)

Contents 8-1
Overview 8-3
Introduction 8-4
How QinQ Works 8-5
Features and Benefits 8-5
Terminology 8-6
Operating Rules and Guidelines 8-7
Enabling QinQ and Configuring QinQ Modes 8-7
QinQ Mixed Vlan Mode 8-8
Configuring VLANs 8-8
Operating Notes and Restrictions 8-10
Configuring QinQ 8-13
General Configuration Steps 8-13
Enabling QinQ 8-14
Setting up S-VLANs 8-14
Configuring Per-Port S-VLAN Membership 8-15

Configuring Port-Types 8-16
Configuration Example 8-17
Updating QinQ Configurations 8-22
Changing QinQ Modes 8-22
Disabling QinQ 8-22
Changing VLAN Port Memberships (Mixed Vlan Mode) 8-22
Moving Ports between C-VLANs and S-VLANs (Mixed Vlan Mode) . 8-23
Displaying QinQ Config and Status 8-24
Show Commands for QinQ 8-24
Show Commands for VLANs 8-25
Displaying Spanning Tree Status 8-27
Effects of QinQ on Other Switch Features 8-28
Event Log Messages and SNMP Support 8-33
SNMP Support and MIB Objects 8-33

9 Classifier-Based Software Configuration

Contents 9-1
Using Classifier-Based Service Policies 9-2
Introduction 9-2
Classifier-Based Configuration Model 9-2
Creating a Traffic Class 9-4
Using Match Criteria 9-4
Class Configuration Procedure 9-5
Optional ICMP Match Criteria 9-13
Optional IGMP Match Criteria 9-16
Optional TCP and UDP Match Criteria 9-17
Using CIDR Notation for IPv4/IPv6 Addresses 9-19
Resequencing Match/Ignore Statements 9-23
Creating a Service Policy 9-24
Modifying Classes in a Policy 9-28
Resequencing Classes in a Policy 9-29
Applying a Service Policy to an Interface 9-30
Where to Go From Here 9-33

Index

Product Documentation

About Your Switch Manual Set

Note

For the latest version of all ProCurve switch documentation, including Release Notes covering recently added features, please visit the ProCurve Networking Web site at **www.procurve.com/manuals**.

Printed Publications

The two publications listed below are printed and shipped with your switch. The latest version of each is also available in PDF format on the ProCurve Web site, as described in the Note at the top of this page.

- *Read Me First*—Provides software update information, product notes, and other information.
- *HP ProCurve Switch Quick Setup*—Provides quick start installation instructions. See the *Installation and Getting Started Guide* for more detailed information.

Electronic Publications

The latest version of each of the publications listed below is available in PDF format on the ProCurve Web site, as described in the Note at the top of this page.

- *Management and Configuration Guide*—Describes how to configure, manage, and monitor basic switch operation.
- *Advanced Traffic Management Guide*—Explains how to configure traffic management features such as VLANs, MSTP, QoS, and Meshing.
- *Multicast and Routing Guide*—Explains how to configure IGMP, PIM, IP routing, and VRRP features.
- *Access Security Guide*—Explains how to configure access security features and user authentication on the switch.
- *IPv6 Configuration Guide*—Describes the IPv6 protocol operations that are supported on the switch.
- *Command Line Interface Reference Guide*—Provides a comprehensive description of CLI commands, syntax, and operations.
- *Event Log Message Reference Guide*—Provides a comprehensive description of event log messages.
- *Release Notes*—Describe new features, fixes, and enhancements that become available between revisions of the main product guide.

Software Feature Index

For the software manual set supporting your 3500/3500yl/5400zl/6200yl/6600/8200zl switch model, this feature index indicates which manual to consult for information on a given software feature.

Note

This Index does not cover IPv6 capable software features. For information on IPv6 protocol operations and features (such as DHCPv6, DNS for IPv6, Ping6, and MLD Snooping), refer to the *IPv6 Configuration Guide*.

Intelligent Edge Software Features. These features are automatically included on all switches.

Premium License Software Features. For the HP ProCurve 3500, 3500yl, 5400zl, 6600, and 8200zl switches, Premium License features can be acquired by purchasing the optional Premium License and installing it on the Intelligent Edge version of these switches. (These features are automatically included on the HP ProCurve 6200yl switches.)

Premium License Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
OSPF			X	
PIM-DM (Dense Mode)			X	
PIM-SM (Sparse Mode)			X	
QinQ (Provider Bridging)		X		
VRRP			X	

Intelligent Edge Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
802.1Q VLAN Tagging		X		
802.1X Port-Based Priority	X			

Intelligent Edge Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
802.1X Multiple Authenticated Clients Per Port				X
Access Control Lists (ACLs)				X
AAA Authentication				X
Authorized IP Managers				X
Authorized Manager List (Web, Telnet, TFTP)				X
Auto MDIX Configuration	X			
BOOTP	X			
Config File	X			
Console Access	X			
Copy Command	X			
Core Dump	X			
CoS (Class of Service)		X		
Debug	X			
DHCP Configuration	X			
DHCP Option 82			X	
DHCP Snooping				X
DHCP/Bootp Operation	X			
Diagnostic Tools	X			
Distributed Trunking	X			
Downloading Software	X			
Dynamic ARP Protection				X
Dynamic Configuration Arbiter				X
Dynamic IP Lockdown				X
Eavesdrop Protection				X
Equal Cost Multi-Path (ECMP)			X	
Event Log	X			

Intelligent Edge Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
Factory Default Settings	X			
Flow Control (802.3x)	X			
File Management	X			
File Transfers	X			
Friendly Port Names	X			
Guaranteed Minimum Bandwidth (GMB)	X			
GVRP		X		
Identity-Driven Management (IDM)		X		
IGMP			X	
Interface Access (Telnet, Console/Serial, Web)	X			
IP Addressing	X			
IP Routing			X	
Jumbo Packets	X			
Key Management System (KMS)				X
LACP	X			
LLDP	X			
LLDP-MED	X			
Loop Protection		X		
MAC Address Management	X			
MAC Lockdown				X
MAC Lockout				X
MAC-based Authentication				X
Management VLAN		X		
Meshing		X		
Monitoring and Analysis	X			
Multicast Filtering				X

Intelligent Edge Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
Multiple Configuration Files	X			
Network Management Applications (SNMP)	X			
Out-of-Band Management (OOBM)	X			
OpenView Device Management	X			
Passwords and Password Clear Protection				X
ProCurve Manager (PCM)	X			
Ping	X			
Port Configuration	X			
Port Monitoring		X		
Port Security				X
Port Status	X			
Port Trunking (LACP)	X			
Port-Based Access Control (802.1X)				X
Power over Ethernet (PoE and PoE+)	X			
Protocol Filters				X
Protocol VLANS		X		
Quality of Service (QoS)		X		
RADIUS Authentication and Accounting				X
RADIUS-Based Configuration				X
Rate-Limiting	X			
RIP			X	
RMON 1,2,3,9	X			
Routing			X	
Routing - IP Static			X	
SavePower Features	X			
Secure Copy	X			

Intelligent Edge Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
sFlow	X			
SFTP	X			
SNMPv3	X			
Software Downloads (SCP/SFTP, TFPT, Xmodem)	X			
Source-Port Filters				X
Spanning Tree (STP, RSTP, MSTP)		X		
SSHv2 (Secure Shell) Encryption				X
SSL (Secure Socket Layer)				X
Stacking (3500/3500yl/6200yl/6600 switches only)		X		
Syslog	X			
System Information	X			
TACACS+ Authentication				X
Telnet Access	X			
TFTP	X			
Time Protocols (TimeP, SNTP)	X			
Traffic Mirroring	X			
Traffic/Security Filters				X
Troubleshooting	X			
Uni-Directional Link Detection (UDLD)	X			
UDP Forwarder			X	
USB Device Support	X			
Virus Throttling (Connection-Rate Filtering)				X
VLANs		X		
VLAN Mirroring (1 static VLAN)		X		
Voice VLAN		X		
Web Authentication RADIUS Support				X

Intelligent Edge Software Features	Manual			
	Management and Configuration	Advanced Traffic Management	Multicast and Routing	Access Security Guide
Web-based Authentication				X
Web UI	X			

1

Getting Started

Contents

Introduction 1-2

Conventions 1-2

- Command Syntax Statements 1-2
- Command Prompts 1-3
- Screen Simulations 1-3
- Configuration and Operation Examples 1-3
- Keys 1-3

Sources for More Information 1-4

- Getting Documentation From the Web 1-6
- Online Help 1-6
  - Menu Interface 1-6
  - Command Line Interface 1-7
  - Web Browser Interface 1-7

Need Only a Quick Start? 1-8

- IP Addressing 1-8

To Set Up and Install the Switch in Your Network 1-8

- Physical Installation 1-8

Introduction

This guide is intended for use with the following ProCurve switches:

- 8200zl switches
- 6600 switches
- 5400zl switches
- 3500, 3500yl and 6200yl switches

It describes how to use the command line interface (CLI), Menu interface, and web browser to configure, manage, monitor, and troubleshoot switch operation. For an overview of product documentation for the above switches, refer to *"Product Documentation"* on page xiii. To download the switch documentation, visit the ProCurve Networking manuals web page at **www.hp.com/go/procurve/manuals**.

Conventions

This guide uses the following conventions for commands and screen displays.

Command Syntax Statements

Syntax: ip < default-gateway < *ip-addr* >> | routing >

Syntax: show interfaces [*port-list*]

- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate optional elements.
- Braces (< >) enclose required elements.
- Braces within square brackets ([< >]) indicate a required element within an optional choice.
- Boldface indicates use of a CLI command, part of a CLI command syntax, or other displayed element in general text. For example:

  "Use the **copy tftp** command to download the key from a TFTP server."

- Italics indicate variables for which you must supply a value when executing the command. For example, in this command syntax, you must provide one or more port numbers:

  Syntax: aaa port-access authenticator < *port-list* >

Command Prompts

In the default configuration, your switch displays a CLI prompt similar to the following example:

```
ProCurve 8212zl#
```

To simplify recognition, this guide uses **ProCurve** to represent command prompts for all switch models. For example:

```
ProCurve#
```

(You can use the **hostname** command to change the text in the CLI prompt.)

Screen Simulations

Displayed Text. Figures containing simulated screen text and command output look like this:

```
ProCurve> show version
Image stamp:   /sw/code/build/info
               March 1, 2007 13:43:13
               K.12.01
               139

ProCurve>
```

Figure 1-1. Example of a Figure Showing a Simulated Screen

In some cases, brief command-output sequences appear without figure identification. For example:

```
ProCurve(config)# clear public-key
ProCurve(config)# show ip client-public-key
show_client_public_key: cannot stat keyfile
```

Configuration and Operation Examples

Unless otherwise noted, examples using a particular switch model apply to all switch models covered by this guide.

Keys

Simulations of actual keys use a bold, sans-serif typeface with square brackets. For example, the Tab key appears as **[Tab]** and the "Y" key appears as **[Y]**.

Sources for More Information

For information about switch operation and features not covered in this guide, consult the following sources:

- Feature Index—For information on which manual to consult for a given software feature, refer to the "Software Feature Index" on page xiv.

Note

For the latest version of all HP ProCurve switch documentation referred to below, including Release Notes covering recently added features, visit the ProCurve Networking manuals web page at **www.hp.com/go/procurve/manuals**.

- Software Release Notes—*Release Notes* are posted on the HP ProCurve Networking web site and provide information on new software updates:
  - new features and how to configure and use them
  - software management, including downloading software to the switch
  - software fixes addressed in current and previous releases
- Product Notes and Software Update Information—The printed *Read Me First* shipped with your switch provides software update information, product notes, and other information.
- *Installation and Getting Started Guide*—Use the *Installation and Getting Started Guide* to prepare for and perform the physical installation. This guide also steps you through connecting the switch to your network and assigning IP addressing, as well as describing the LED indications for correct operation and trouble analysis.
- *Management and Configuration Guide*—Use this guide for information on topics such as:
  - various interfaces available on the switch
  - memory and configuration operation
  - interface access
  - IP addressing

- time protocols
- port configuration, trunking, traffic control, and PoE operation
- Redundant management
- SNMP, LLDP, and other network management topics
- file transfers, switch monitoring, troubleshooting, and MAC address management

■ *Advanced Traffic Management Guide*—Use this guide for information on topics such as:

- VLANs: Static port-based and protocol VLANs, and dynamic GVRP VLANs
- spanning-Tree: 802.1D (STP), 802.1w (RSTP), and 802.1s (MSTP)
- meshing
- Quality-of-Service (QoS)
- Access Control Lists (ACLs)
- Out-of-Band Management (6600)

■ *Multicast and Routing Guide*—Use this guide for information on topics such as:

- IGMP
- PIM (SM and DM)
- IP routing
- VRRP

■ *Access Security Guide*—Use this guide for information on topics such as:

- Local username and password security
- Web-Based and MAC-based authentication
- RADIUS and TACACS+ authentication
- SSH (Secure Shell) and SSL (Secure Socket Layer) operation
- 802.1X access control
- Port security operation with MAC-based control
- Authorized IP Manager security
- Key Management System (KMS)

■ *IPv6 Configuration Guide*—Use this guide for information on topics such as:

- Overview of IPv6 operation and features supported in software release K.13.01 or greater
- Configuring IPv6 addressing
- Using IPv6 management, security, and troubleshooting features

Getting Documentation From the Web

To obtain the latest versions of documentation and release notes for your switch, go to the ProCurve Networking manuals web page at **www.hp.com/go/procurve/manuals.**

Online Help

Menu Interface

If you need information on specific parameters in the menu interface, refer to the online help provided in the interface. For example:

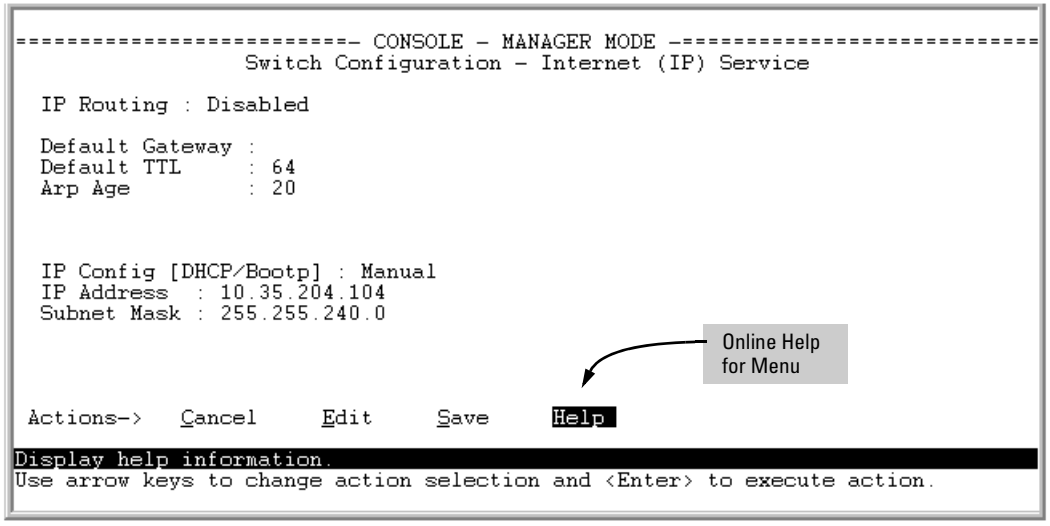

========================== CONSOLE - MANAGER MODE ==========================
Switch Configuration - Internet (IP) Service
IP Routing : Disabled
Default Gateway :
Default TTL : 64
Arp Age : 20
IP Config [DHCP/Bootp] : Manual
IP Address : 10.35.204.104
Subnet Mask : 255.255.240.0
Online Help for Menu
Actions-> Cancel Edit Save Help
Display help information.
Use arrow keys to change action selection and <Enter> to execute action.

Figure 1-2. Online Help for Menu Interface

Command Line Interface

If you need information on a specific command in the CLI, type the command name followed by **help**. For example:

```
ProCurve# write help
Usage: write <memory|terminal>

Description: View or save the running configuration of the switch.
             write terminal - displays the running configuration of the
                              switch on the terminal
             write memory   - saves the running configuration of the
                              switch to flash. The saved configuration
                              becomes the boot-up configuration of the switch
                              the next time it is booted.
```

Figure 1-3. Example of CLI Help

WebAgent (Web Browser Interface)

If you need information on specific features in the HP ProCurve WebAgent, use the online Help. You can access the Help by clicking on the "**Help**" text in any WebAgent screen.

To download the WebAgent help files to a local server, go to:

www.hp.com/rnd/device_help/download.htm

Follow the directions on the web page to download the WebAgent help for your device.

Need Only a Quick Start?

IP Addressing

If you just want to give the switch an IP address so that it can communicate on your network, or if you are not using VLANs, ProCurve recommends that you use the Switch Setup screen to quickly configure IP addressing. To do so, do one of the following:

- Enter **setup** at the CLI Manager level prompt.

```
Procurve# setup
```

- In the Main Menu of the Menu interface select

  8. Run Setup

For more on using the Switch Setup screen, see the *Installation and Getting Started Guide* you received with the switch.

To Set Up and Install the Switch in Your Network

Physical Installation

Use the *Installation and Getting Started Guide* for the following:

- Notes, cautions, and warnings related to installing and using the switch and its related modules
- Instructions for physically installing the switch in your network
- Quickly assigning an IP address and subnet mask, set a Manager password, and (optionally) configure other basic features.
- Interpreting LED behavior.

For the latest version of the *Installation and Getting Started Guide* for your switch, refer to “Getting Documentation From the Web” on page 1-6.

2

Static Virtual LANs (VLANs)

Contents

Overview 2-4

Introduction 2-5

General VLAN Operation 2-5

Types of Static VLANs Available in the Switch 2-6

Port-Based VLANs 2-6

Protocol-Based VLANs 2-6

Designated VLANs 2-6

Terminology 2-7

Static VLAN Operation 2-8

VLAN Environments 2-9

VLAN Operation 2-10

Routing Options for VLANs 2-11

Overlapping (Tagged) VLANs 2-12

Per-Port Static VLAN Configuration Options 2-14

VLAN Operating Rules 2-15

General Steps for Using VLANs 2-19

Multiple VLAN Considerations 2-20

Single Forwarding Database Operation 2-21

Example of an Unsupported Configuration and How To Correct It 2-22

Multiple Forwarding Database Operation 2-23

Configuring VLANs 2-24

Menu: Configuring Port-Based VLAN Parameters 2-24

To Change VLAN Support Settings 2-25

Adding or Editing VLAN Names 2-26

Adding or Changing a VLAN Port Assignment 2-28

CLI: Configuring Port-Based and Protocol-Based VLAN Parameters 2-30

Customizing the Show VLANs Output 2-36
Creating an Alias for Show VLAN Commands 2-38
Note on Using Pattern Matching with the "Show VLANs Custom" Command 2-39
Changing the Number of VLANs Allowed on the Switch 2-39
Web: Viewing and Configuring VLAN Parameters 2-45

802.1Q VLAN Tagging 2-46

Special VLAN Types 2-51
VLAN Support and the Default VLAN 2-51
The Primary VLAN 2-51
The Secure Management VLAN 2-52
Preparation 2-54
Configuration 2-55
Using DHCP to Obtain an IP Address 2-56
Deleting the Management VLAN 2-59
Operating Notes for Management VLANs 2-59
Voice VLANs 2-60
Operating Rules for Voice VLANs 2-60
Components of Voice VLAN Operation 2-61
Voice VLAN QoS Prioritizing (Optional) 2-61
Voice VLAN Access Security 2-62

Effect of VLANs on Other Switch Features 2-62
Spanning Tree Operation with VLANs 2-62
IP Interfaces 2-63
VLAN MAC Address 2-63
Port Trunks 2-63
Port Monitoring 2-63
Jumbo Packet Support 2-63

VLAN Restrictions 2-64

Migrating Layer 3 VLANs Using VLAN MAC Configuration 2-65
VLAN MAC Address Reconfiguration 2-65
Handling Incoming and Outgoing VLAN Traffic 2-66
Sending Heartbeat Packets with a Configured MAC Address 2-67
Configuring a VLAN MAC Address with Heartbeat Interval 2-68
Operating Notes 2-68

Example 2-69
Verifying a VLAN MAC Address Configuration 2-69

Overview

This chapter describes how to configure and use static, port-based and protocol-based VLANs on the switches covered in this guide.

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

- Chapter 3, "Using the Menu Interface"
- Chapter 4, "Using the Command Line Interface (CLI)"
- Chapter 5, "Using the Web Browser Interface
- Chapter 6, "Switch Memory and Configuration"

Introduction

VLAN Features

Feature	Default	Menu	CLI	Web
view existing VLANs	n/a	page 2-25 thru 2-30	page 2-31	page 2-45
configuring static VLANs	default VLAN with VID = 1	page 2-25 thru 2-30	page 2-30	page 2-45

VLANs enable you to group users by logical function instead of physical location. This helps to control bandwidth usage within your network by allowing you to group high-bandwidth users on low-traffic segments and to organize users from different LAN segments according to their need for common resources and/or their use of individual protocols. You can also improve traffic control at the edge of your network by separating traffic of different protocol types. VLANs can also enhance your network security by creating separate subnets to help control in-band access to specific network resources.

General VLAN Operation

A VLAN is comprised of multiple ports operating as members of the same subnet (broadcast domain). Ports on multiple devices can belong to the same VLAN, and traffic moving between ports in the same VLAN is bridged (or "switched"). (Traffic moving between different VLANs must be routed.) A s*tatic* VLAN is an 802.1Q-compliant VLAN configured with one or more ports that remain members regardless of traffic usage. (A *dynamic* VLAN is an 802.1Q-compliant VLAN membership that the switch temporarily creates on a port to provide a link to another port in the same VLAN on another device.)

This chapter describes *static* VLANs configured for port-based or protocol-based operation. Static VLANs are configured with a name, VLAN ID number (VID), and port members. (For *dynamic* VLANs, refer to chapter 3, "GVRP" .)

By default, the switches covered in this guide are 802.1Q VLAN-enabled and allow up to 2048 static and dynamic VLANs. (The default static VLAN setting is 8). 802.1Q compatibility enables you to assign each switch port to multiple VLANs, if needed.

Types of Static VLANs Available in the Switch

Port-Based VLANs

This type of static VLAN creates a specific layer-2 broadcast domain comprised of member ports that bridge IPv4 traffic among themselves. Port-Based VLAN traffic is routable on the switches covered in this guide.

Protocol-Based VLANs

This type of static VLAN creates a layer-3 broadcast domain for traffic of a particular protocol, and is comprised of member ports that bridge traffic of the specified protocol type among themselves. Some protocol types are routable on the switches covered in this guide. Refer to table 2-1 on page 2-8.

Designated VLANs

The switch uses these static, port-based VLAN types to separate switch management traffic from other network traffic. While these VLANs are not limited to management traffic only, they can provide improved security and availability for management traffic.

- **The Default VLAN:** This port-based VLAN is always present in the switch and, in the default configuration, includes all ports as members (page 2-51).
- **The Primary VLAN:** The switch uses this port-based VLAN to run certain features and management functions, including DHCP/Bootp responses for switch management. In the default configuration, the Default VLAN is also the Primary VLAN. However, you can designate another, port-based, non-default VLAN, as the Primary VLAN (page 2-51).
- **The Secure Management VLAN:** This optional, port-based VLAN establishes an isolated network for managing the ProCurve switches that support this feature. Access to this VLAN and to the switch's management functions are available only through ports configured as members (page 2-52).
- **Voice VLANs:** This optional, port-based VLAN type enables you to separate, prioritize, and authenticate voice traffic moving through your network, and to avoid the possibility of broadcast storms affecting VoIP (Voice-over-IP) operation (page 2-60).

Note

In a multiple-VLAN environment that includes some older switch models there may be problems related to the same MAC address appearing on different ports and VLANs on the same switch. In such cases the solution is to impose some cabling and VLAN restrictions. For more on this topic, refer to "Multiple VLAN Considerations" on page 2-20.

Terminology

Dynamic VLAN: An 802.1Q VLAN membership temporarily created on a port linked to another device, where both devices are running GVRP. (See also **Static VLAN**.) For more information, refer to chapter 3, "GVRP" .

Static VLAN: A port-based or protocol-based VLAN configured in switch memory. (See also **Dynamic VLAN**.)

Tagged Packet: A packet that carries an IEEE 802.1Q VLAN ID (VID), which is a two-byte extension that precedes the source MAC address field of an ethernet frame. A VLAN tag is layer 2 data and is transparent to higher layers.

Tagged VLAN: A VLAN that complies with the 802.1Q standard, including priority settings, and allows a port to join multiple VLANs. (See also **Untagged VLAN**.)

Untagged Packet: A packet that does not carry an IEEE 802.1Q VLAN ID (VID).

Untagged VLAN: A VLAN that does not use or forward 802.1Q VLAN tagging, including priority settings. A port can be a member of only one untagged VLAN of a given type (port-based and the various protocol-based types). (See also **Tagged VLAN**.)

VID: The acronym for a VLAN Identification Number. Each 802.1Q-compliant VLAN must have its own unique VID number, and that VLAN must be given the same VID in every device in which it is configured.

Static VLAN Operation

A group of networked ports assigned to a VLAN form a broadcast domain that is separate from other VLANs that may be configured on the switch. On a given switch, packets are bridged between source and destination ports that belong to the same VLAN. Thus, all ports passing traffic for a particular subnet address should be configured to the same VLAN. Cross-domain broadcast traffic in the switch is eliminated and bandwidth is saved by not allowing packets to flood out all ports.

Table 2-1. Comparative Operation of Port-Based and Protocol-Based VLANs

	Port-Based VLANs	Protocol-Based VLANs
IP Addressing	Usually configured with at least one unique IP address. You can create a port-based VLAN without an IP address. However, this limits the switch features available to ports on that VLAN. (Refer to "How IP Addressing Affects Switch Operation" in the chapter "Configuring IP Addressing" in the *Management and Configuration Guide* for the switch.) You can also use multiple IP addresses to create multiple subnets within the same VLAN. (For more on this topic, refer to the chapter on "Configuring IP Addressing" in the *Management and Configuration Guide* for the switch.)	You can configure IP addresses on all protocol VLANs. However, IP addressing is used only on IPv4 and IPv6 protocol VLANs. **Restrictions**: When you configure an IP address on a VLAN interface, the following restrictions apply: Loopback interfaces share the same IP address space with VLAN configurations. The maximum number of IP addresses supported on a switch is 2048, which includes all IP addresses configured for both VLANs and loopback interfaces (except for the default loopback IP address 127.0.0.1). Each IP address that you configure on a VLAN interface must be unique in the switch. This means that the address cannot be used by a VLAN interface or another loopback interface. For more information, refer to the chapter on "Configuring IP Addressing" in the *Management and Configuration Guide*.

	Port-Based VLANs	Protocol-Based VLANs
Untagged VLAN Membership	A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol VLAN of a specific protocol type (such as IPX or IPv6). If the same protocol type is configured in multiple protocol VLANs, then a port can be an untagged member of only one of those protocol VLANs. For example, if you have two protocol VLANs, 100 and 200, and both include IPX, then a port can be an untagged member of either VLAN 100 or VLAN 200, but not both VLANs. A port's untagged VLAN memberships can include up to four different protocol types. This means that a port can be an untagged member of one of the following: • Four single-protocol VLANs • Two protocol VLANs where one VLAN includes a single protocol and the other includes up to three protocols • One protocol VLAN where the VLAN includes four protocols
Tagged VLAN Membership	A port can be a tagged member of any port-based VLAN. See above.	A port can be a tagged member of any protocol-based VLAN. See above.
Routing	The switch can internally route IP (IPv4) traffic between port-based VLANs and between port-based and IPv4 protocol-based VLANs if the switch configuration enables IP routing. If the switch is not configured to route traffic internally between port-based VLANs, then an external router must be used to move traffic between VLANs.	If the switch configuration enables IP routing, the switch can internally route IPv4 traffic as follows: • Between multiple IPv4 protocol-based VLANs • Between IPv4 protocol-based VLANs and port-based VLANs. Other protocol-based VLANs require an external router for moving traffic between VLANs. **Note:** NETbeui and SNA are non-routable protocols. End stations intended to receive traffic in these protocols must be attached to the same physical network.
Commands for Configuring Static VLANs	vlan < ***VID*** > [tagged | untagged < [e] ***port-list*** >]	vlan < *VID* > protocol < ipx | ipv4 | ipv6 | arp | appletalk | sna | netbeui > vlan < *VID* > [tagged | untagged < [e] ***port-list*** >]

VLAN Environments

You can configure different VLAN types in any combination. Note that the default VLAN will always be present. (For more on the default VLAN, refer to "VLAN Support and the Default VLAN" on page 2-51.)

Table 2-2. VLAN Environments

VLAN Environment	Elements
The default VLAN (port-based; VID of "1") Only	In the default VLAN configuration, all ports belong to VLAN 1 as untagged members. VLAN 1 is a port-based VLAN, for IPv4 traffic.
Multiple VLAN Environment	In addition to the default VLAN, the configuration can include one or more other port-based VLANs and one or more protocol VLANs. (The switches covered in this guide allow up to 2048 (vids up to 4094) VLANs of all types.) Using VLAN tagging, ports can belong to multiple VLANs of all types. Enabling routing on the switch enables the switch to route IPv4 traffic between port-based VLANs and between port-based VLANs and IPv4 protocol VLANs. Routing other types of traffic between VLANs requires an external router capable of processing the appropriate protocol(s).

VLAN Operation

The Default VLAN. In figure 2-1, all ports belong to the default VLAN, and devices connected to these ports are in the same broadcast domain. Except for an IP address and subnet, no configuration steps are needed.

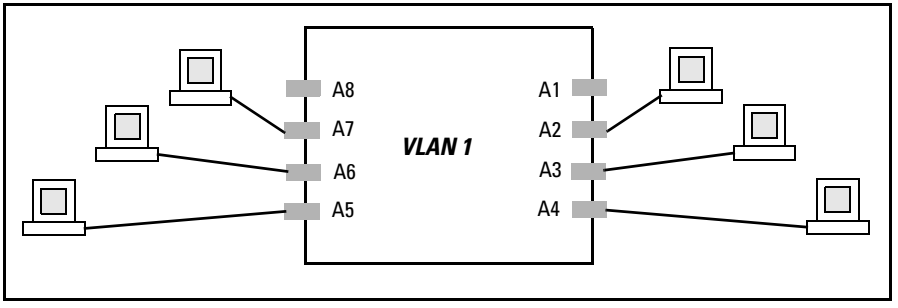


Figure 2-1. Example of a Switch in the Default VLAN Configuration

Multiple Port-Based VLANs. In figure 2-2, routing within the switch is disabled (the default). This means that communication between any routable VLANs on the switch must go through the external router. In this case, VLANs "W" and "X" can exchange traffic through the external router, but traffic in VLANs "Y" and "Z" is restricted to the respective VLANs. Note that VLAN 1, the default VLAN, is also present, but not shown. (The default VLAN cannot be deleted from the switch. However, ports assigned to other VLANs can be removed from the default VLAN, if desired.) If internal (IP) routing is enabled

on the switch, then the external router is not needed for traffic to move between port-based VLANs.

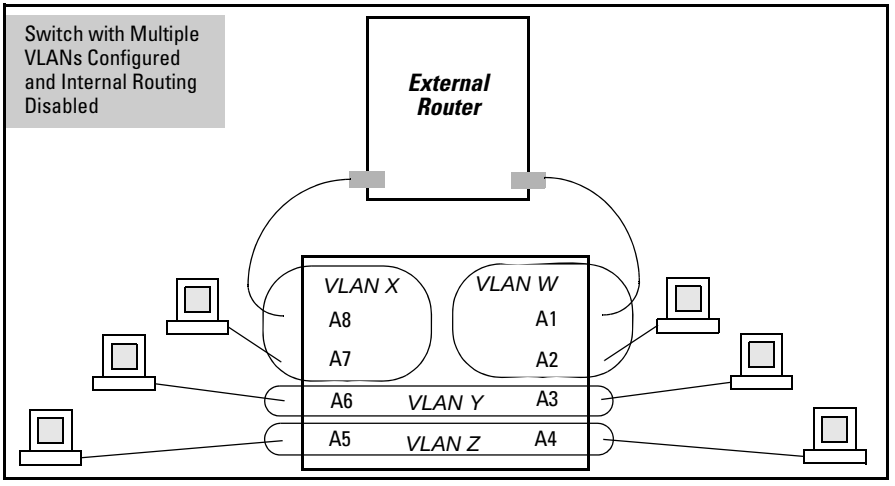


Figure 2-2. Example of Multiple VLANs on the Switch

Protocol VLAN Environment. Figure 2-2 can also be applied to a protocol VLAN environment. In this case, VLANs "W" and "X" represent routable protocol VLANs. VLANs "Y" and "Z" can be any protocol VLAN. As noted for the discussion of multiple port-based VLANs, VLAN 1 is not shown. Enabling internal (IP) routing on the switch allows IP traffic to move between VLANs on the switch. However, routable, non-IP traffic always requires an external router.

Routing Options for VLANs

Table 2-3. Options for Routing Between VLAN Types in the Switch

	Port-Based	IPX	IPv4	IPv6	ARP	Apple-Talk	SNA[2]	Netbeui[2]
Port-Based	Yes	—	Yes	—	—	—	—	—
Protocol								
IPX	—	Yes[1]	—	—	—	—	—	—
IPv4	Yes	—	Yes	—	—	—	—	—
IPv6	—	—	—	Yes[1]	—	—	—	—
ARP	—	—	—	—	Yes[1]	—	—	—
AppleTalk	—	—	—	—	—	Yes[1]	—	—

	Port-Based	IPX	IPv4	IPv6	ARP	Apple-Talk	SNA[2]	Netbeui[2]
SNA[2]	—	—	—	—	—	—	—	—
NETbeui[2]	—	—	—	—	—	—	—	—

[1]Requires an external router to route between VLANs.

[2]Not a routable protocol type. End stations intended to receive traffic in these protocols must be attached to the same physical network.

Overlapping (Tagged) VLANs

A port can be a member of more than one VLAN of the same type if the device to which the port connects complies with the 802.1Q VLAN standard. For example, a port connected to a central server using a network interface card (NIC) that complies with the 802.1Q standard can be a member of multiple VLANs, allowing members of multiple VLANs to use the server. Although these VLANs cannot communicate with each other through the server, they can all access the server over the same connection from the switch. Where VLANs overlap in this way, VLAN "tags" are used in the individual packets to distinguish between traffic from different VLANs. A VLAN tag includes the particular VLAN I.D. (VID) of the VLAN on which the packet was generated.

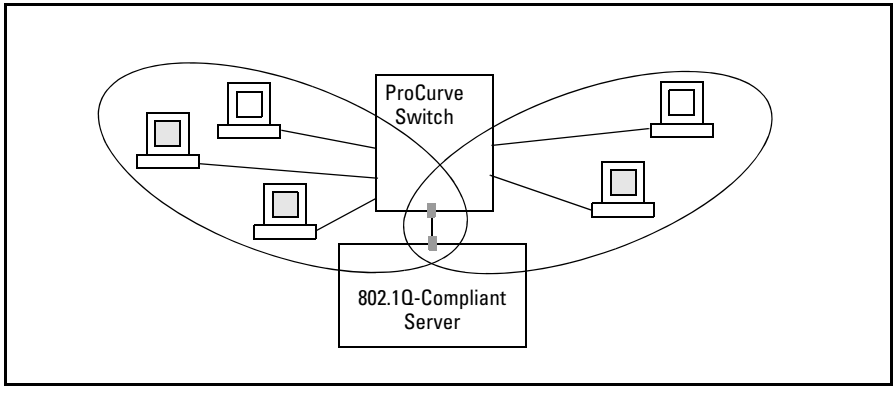


Figure 2-3. Example of Overlapping VLANs Using the Same Server

Similarly, using 802.1Q-compliant switches, you can connect multiple VLANs through a single switch-to-switch link.

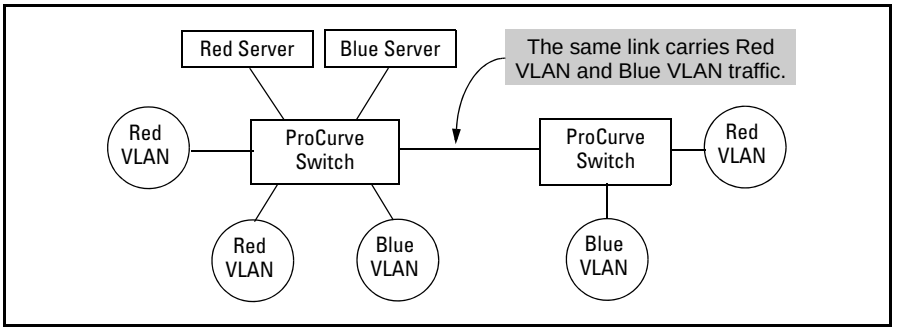


Figure 2-4. Example of Connecting Multiple VLANs Through the Same Link

Introducing Tagged VLAN Technology into Networks Running Legacy (Untagged) VLANs. You can introduce 802.1Q-compliant devices into networks that have built untagged VLANs based on earlier VLAN technology. The fundamental rule is that legacy/untagged VLANs require a separate link for each VLAN, while 802.1Q, or tagged VLANs can combine several VLANs in one link. This means that on the 802.1Q-compliant device, separate ports (configured as untagged) must be used to connect separate VLANs to non-802.1Q devices.

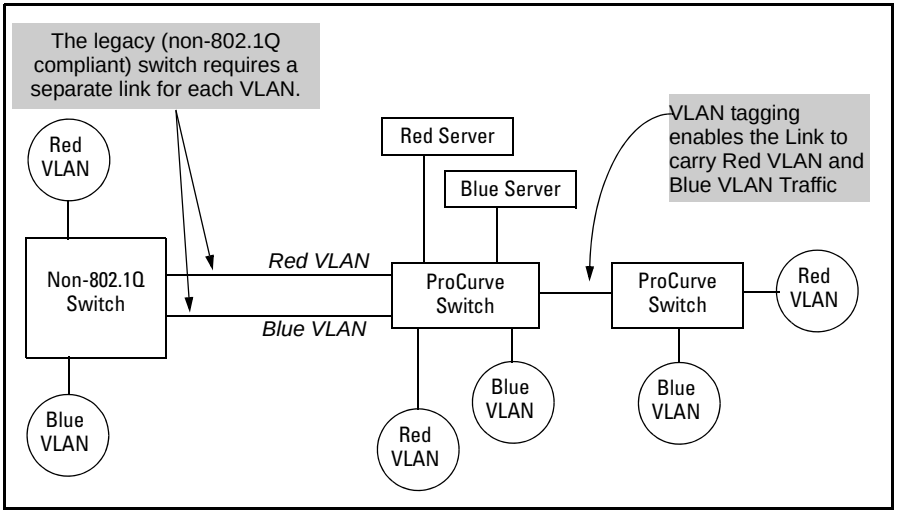


Figure 2-5. Example of Tagged and Untagged VLAN Technology in the Same Network

For more information on VLANs, refer to:

- "Overview of Using VLANs" (page 2-51)
- "Menu: Configuring VLAN Parameters (page 2-24)

- "CLI: Configuring VLAN Parameters" (page 2-24)
- "Web: Viewing and Configuring VLAN Parameters" (page 2-45)
- "VLAN Tagging Information" (page 2-46)
- "Effect of VLANs on Other Switch Features" (page 2-62)
- "VLAN Restrictions" (page 2-64)

Per-Port Static VLAN Configuration Options

The following figure and table show the options you can use to assign individual ports to a static VLAN. Note that GVRP, if configured, affects these options and VLAN behavior on the switch. The display below shows the per-port VLAN configuration options. Table 2-4 briefly describes these options.

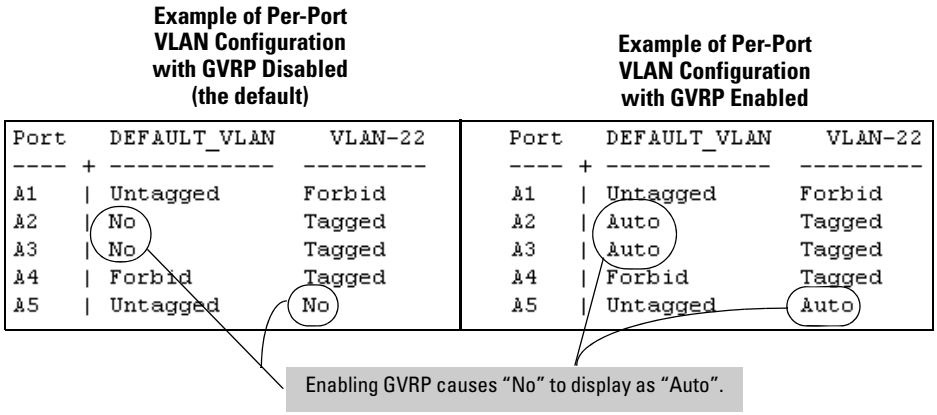

Example of Per-Port VLAN Configuration with GVRP Disabled (the default)
Example of Per-Port VLAN Configuration with GVRP Enabled
Port DEFAULT_VLAN VLAN-22
A1 Untagged Forbid
A2 No Tagged
A3 No Tagged
A4 Forbid Tagged
A5 Untagged No
Port DEFAULT_VLAN VLAN-22
A1 Untagged Forbid
A2 Auto Tagged
A3 Auto Tagged
A4 Forbid Tagged
A5 Untagged Auto
Enabling GVRP causes "No" to display as "Auto".

Figure 2-6. Comparing Per-Port VLAN Options With and Without GVRP

Table 2-4. Per-Port VLAN Configuration Options

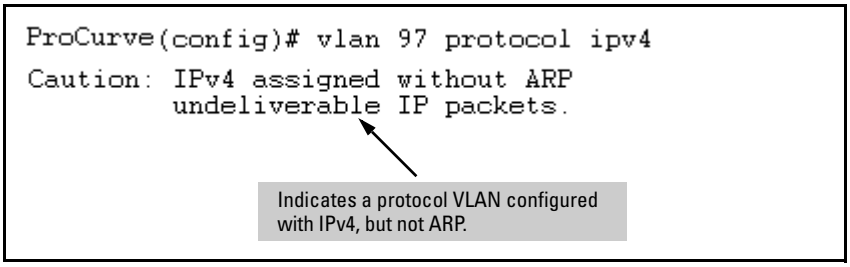
Parameter	Effect on Port Participation in Designated VLAN
Tagged	Allows the port to join multiple VLANs.
Untagged	Allows VLAN connection to a device that is configured for an untagged VLAN instead of a tagged VLAN. A port can be an untagged member of only one port-based VLAN. A port can also be an untagged member of only one protocol-based VLAN for any given protocol type. For example, if the switch is configured with the default VLAN plus three protocol-based VLANs that include IPX, then port 1 can be an untagged member of the default VLAN and one of the protocol-based VLANS.

Parameter	Effect on Port Participation in Designated VLAN
No **- or -** **Auto**	**No**: Appears when the switch is not GVRP-enabled; prevents the port from joining that VLAN. **Auto**: Appears when GVRP is enabled on the switch; allows the port to dynamically join any advertised VLAN that has the same VID
Forbid	Prevents the port from joining the VLAN, even if GVRP is enabled on the switch.

VLAN Operating Rules

- **DHCP/Bootp:** If you are using DHCP/Bootp to acquire the switch's configuration, packet time-to-live, and TimeP information, you must designate the VLAN on which DHCP is configured for this purpose as the Primary VLAN. (In the factory-default configuration, the DEFAULT_VLAN is the Primary VLAN.)
- **Per-VLAN Features:** IGMP and some other features operate on a "per VLAN" basis. This means you must configure such features separately for each VLAN in which you want them to operate.
- **Default VLAN:** You can rename the default VLAN, but you cannot change its VID (1) or delete it from the switch.
- **VLAN Port Assignments:** Any ports *not* specifically removed from the default VLAN remain in the DEFAULT_VLAN, regardless of other port assignments. Also, a port must always be a tagged or untagged member of at least one port-based VLAN.
- **Voice-Over-IP (VoIP):** VoIP operates only over static, port-based VLANs.
- **Multiple VLAN Types Configured on the Same Port:** A port can simultaneously belong to both port-based and protocol-based VLANs.
- **Protocol Capacity:** A protocol-based VLAN can include up to four protocol types. In protocol VLANs using the IPv4 protocol, ARP must be one of these protocol types (to support normal IP network operation). Otherwise, IP traffic on the VLAN is disabled. If you configure an IPv4

protocol VLAN that does not already include the ARP VLAN protocol, the switch displays this message:


ProCurve(config)# vlan 97 protocol ipv4
Caution: IPv4 assigned without ARP
undeliverable IP packets.
Indicates a protocol VLAN configured with IPv4, but not ARP.

- **Deleting Static VLANs:** On the switches covered in this guide you can delete a VLAN regardless of whether there are currently any ports belonging to that VLAN. (The ports are moved to the default VLAN.)
- **Adding or Deleting VLANs:** Changing the number of VLANs supported on the switch requires a reboot. (From the CLI, you must perform a **write memory** command before rebooting.) Other VLAN configuration changes are dynamic.
- **Inbound Tagged Packets:** If a tagged packet arrives on a port that is not a tagged member of the VLAN indicated by the packet's VID, the switch drops the packet. Similarly, the switch will drop an inbound, tagged packet if the receiving port is an *untagged* member of the VLAN indicated by the packet's VID.
- **Untagged Packet Forwarding:** To enable an inbound port to forward an untagged packet, the port must be an untagged member of either a protocol VLAN matching the packet's protocol or an untagged member of a port-based VLAN. That is, when a port receives an incoming, untagged packet, it processes the packet according to the following ordered criteria:
  a. If the port has no untagged VLAN memberships, the switch drops the packet.
  b. If the port has an untagged VLAN membership in a protocol VLAN that matches the protocol type of the incoming packet, then the switch forwards the packet on that VLAN.
  c. If the port is a member of an untagged, port-based VLAN, the switch forwards the packet to that VLAN. Otherwise, the switch drops the packet.

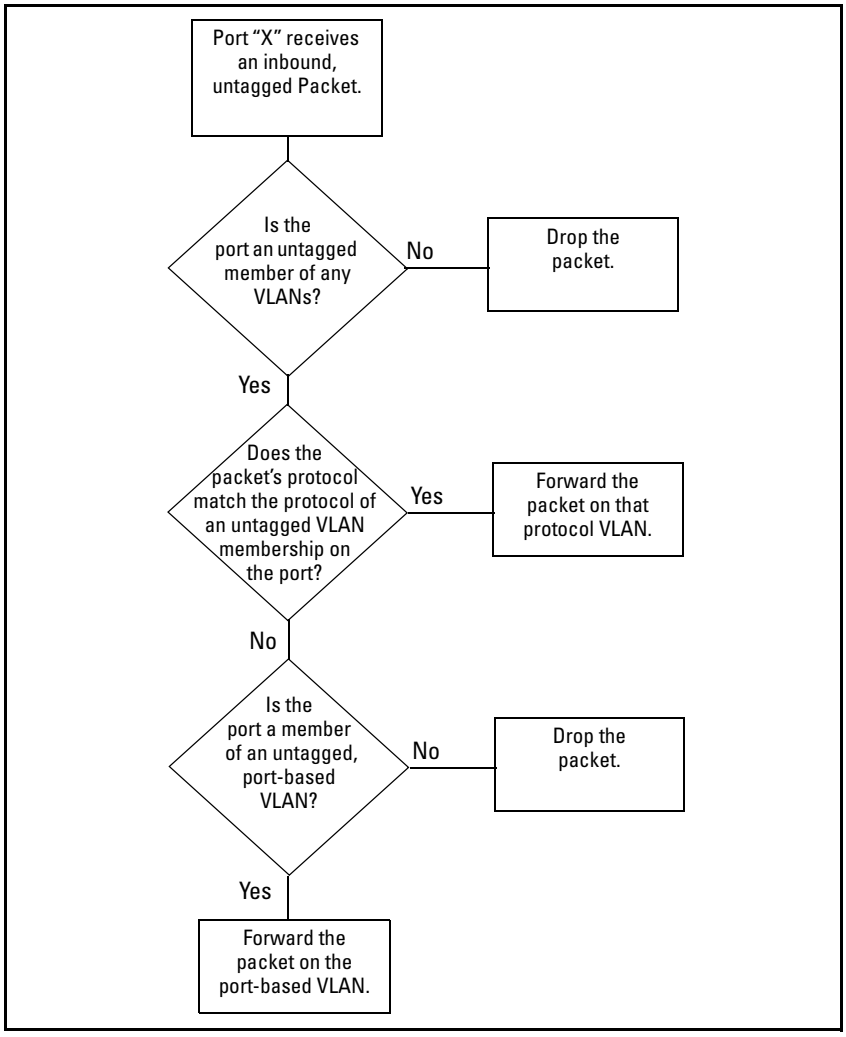

Port "X" receives an inbound, untagged Packet.
Is the port an untagged member of any VLANs?
No
Drop the packet.
Yes
Does the packet's protocol match the protocol of an untagged VLAN membership on the port?
Yes
Forward the packet on that protocol VLAN.
No
Is the port a member of an untagged, port-based VLAN?
No
Drop the packet.
Yes
Forward the packet on the port-based VLAN.

Figure 2-7. Untagged VLAN Operation

- **Tagged Packet Forwarding:** If a port is a tagged member of the same VLAN as an inbound, tagged packet received on that port, then the switch forwards the packet to an outbound port on that VLAN. (To enable the forwarding of tagged packets, any VLAN to which the port belongs as a

tagged member must have the same VID as that carried by the inbound, tagged packets generated on that VLAN.)

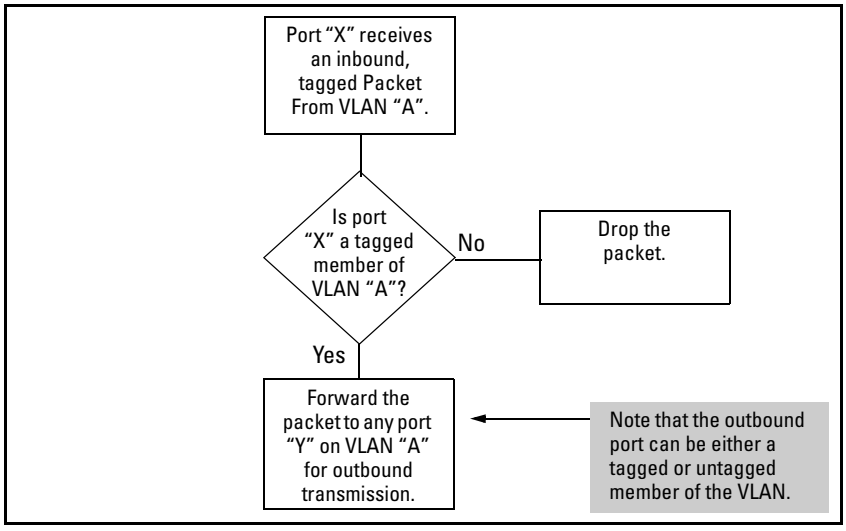

Port "X" receives an inbound, tagged Packet From VLAN "A".
Is port "X" a tagged member of VLAN "A"?
No
Drop the packet.
Yes
Forward the packet to any port "Y" on VLAN "A" for outbound transmission.
Note that the outbound port can be either a tagged or untagged member of the VLAN.

Figure 2-8. Tagged VLAN Operation

See also "Multiple VLAN Considerations" on page 2-20.

Caution

Rate-limiting may behave unpredictably on a VLAN if the VLAN spans multiple modules or port-banks. This also applies if a port on a different module or port-bank is added to an existing VLAN. ProCurve does not recommend configuring rate-limiting on VLANs that include ports spanning modules or port-banks.

In figure 2-9 ports 2, 3, and 24 form one VLAN. The ports are in the same port-bank, which includes ports 1 through 24. Ports 28, 29, and 32 form a second VLAN. These ports are also in the same port-bank, which includes ports 25 through 48. Rate-limiting will operate as expected for these VLANs.

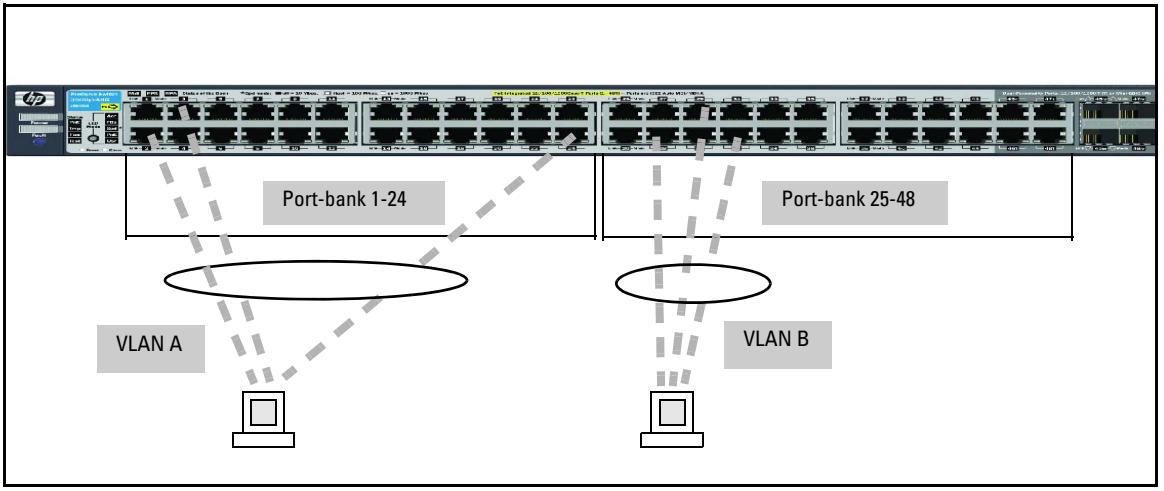


Figure 2-9. Example of VLANs Using Ports from the Same Port-Bank for Each VLAN

General Steps for Using VLANs

1. Plan your VLAN strategy and create a map of the logical topology that will result from configuring VLANs. Include consideration for the interaction between VLANs and other features such as Spanning Tree Protocol, port trunking, and IGMP. (Refer to "Effect of VLANs on Other Switch Features" on page 2-62.) If you plan on using dynamic VLANs, include the port configuration planning necessary to support this feature. (Refer to chapter 3, "GVRP" .)

   By default, VLAN support is enabled for up to 256 VLANs.

2. Configure at least one VLAN in addition to the default VLAN.
3. Assign the desired switch ports to the new VLAN(s).
4. If you are managing VLANs with SNMP in an IP network, the VLAN through which you are managing the switch must have an IP address. For information on the procedure and restrictions when you configure an IP address on a VLAN interface, refer to Table 2-1 on page 2-8.

Multiple VLAN Considerations

Switches use a *forwarding database* to maintain awareness of which external devices are located on which VLANs. Some switches, such as the switches covered in this guide, have a *multiple forwarding database*, which means the switch allows multiple database entries of the same MAC address, with each entry showing the (different) source VLAN and source port. Other switch models have a *single forwarding database*, which means they allow only one database entry of a unique MAC address, along with the source VLAN and source port on which it is found. All VLANs on a switch use the same MAC address. Thus, connecting a multiple forwarding database switch to a single forwarding database switch where multiple VLANs exist imposes some cabling and port VLAN assignment restrictions. Table 2-5 illustrates the functional difference between the two database types.

Table 2-5. Example of Forwarding Database Content

Multiple Forwarding Database		
MAC Address	**Destination VLAN ID**	**Destination Port**
0004ea-84d9f4	1	A5
0004ea-84d9f4	22	A12
0004ea-84d9f4	44	A20
0060b0-880a81	33	A20

This database allows multiple destinations for the same MAC address. If the switch detects a new destination for an existing MAC entry, it just **adds** a new instance of that MAC to the table.

Single Forwarding Database		
MAC Address	**Destination VLAN ID**	**Destination Port**
0004ea-84d9f4	100	A9
0060b0-880af9	105	A10
0060b0-880a81	107	A17

This database allows only one destination for a MAC address. If the switch detects a new destination for an existing MAC entry, it **replaces** the existing MAC instance with a new instance showing the new destination.

Table 2-6 lists the database structure of current ProCurve switch models.

Table 2-6. Forwarding Database Structure for Managed ProCurve Switches

Multiple Forwarding Databases*	Single Forwarding Database*
Series 8200zl switches	Switch 1600M/2400M/2424M
Switch 6600	Switch 4000M/8000M
Series 6400cl switches	Series 2500 switches
Switch 6200yl	Switch 2000
Switch 6108	Switch 800T
Series 5400zl switches	
Series 5300xl switches	
Series 4200vl switches	
Series 4100gl switches	
Series 3500 switches	
Series 3500yl switches	
Series 3400cl switches	
Switch 2810	
Series 2800 switches	
Series 2600/2600-PWR switches	
Series 2510 switches	

*To determine whether other vendors' devices use single-forwarding or multiple-forwarding database architectures, refer to the documentation provided for those devices.

Single Forwarding Database Operation

When a packet arrives with a destination MAC address that matches a MAC address in the switch's forwarding table, the switch tries to send the packet to the port listed for that MAC address. But, if the destination port is in a different VLAN than the VLAN on which the packet was received, the switch drops the packet. This is not a problem for a switch with a multiple forwarding database (refer to table 2-6, above) because the switch allows multiple instances of a given MAC address; one for each valid destination. However, a switch with a single forwarding database allows only one instance of a given MAC address. If (1) you connect the two types of switches through multiple ports or trunks belonging to different VLANs, and (2) enable routing on the switch having the multiple forwarding database; then, on the switch having the single forwarding database, the port and VLAN record it maintains for the connected multiple-forwarding-database switch can frequently change. This causes poor performance and the appearance of an intermittent or broken connection.

Example of an Unsupported Configuration and How To Correct It

The Problem. In figure 2-10, the MAC address table for Switch 8000M will sometimes record the switch as accessed on port A1 (VLAN 1), and other times as accessed on port B1 (VLAN 2):

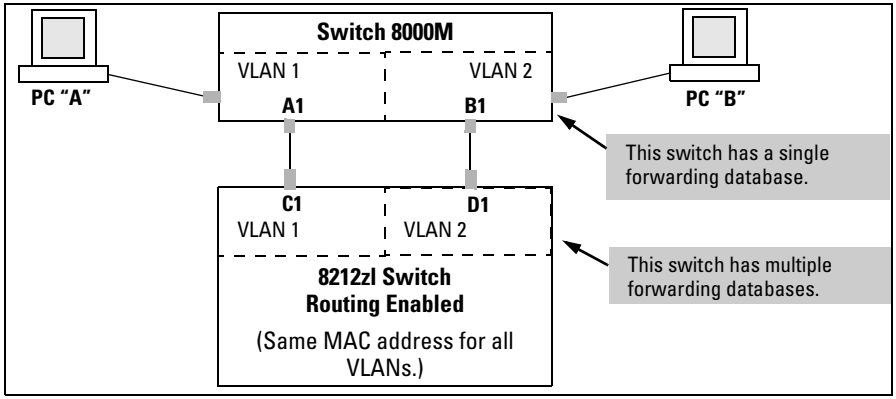


Figure 2-10. Example of Invalid Configuration for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

In figure 2-10, PC "A" sends an IP packet to PC "B".

1. The packet enters VLAN 1 in the Switch 8000 with the 8212zl switch's MAC address in the destination field. Because the 8000M has not yet learned this MAC address, it does not find the address in its address table, and floods the packet out all ports, including the VLAN 1 link (port "A1") to the 8212zl switch. The 8212zl switch then routes the packet through the VLAN 2 link to the 8000M, which forwards the packet on to PC "B". Because the 8000M received the packet from the 8212zl switch on VLAN 2 (port "B1"), the 8000M's single forwarding database records the 8212zl switch as being on port "B1" (VLAN 2).
2. PC "A" now sends a second packet to PC "B". The packet again enters VLAN 1 in the Switch 8000 with the 8212zl switch's MAC address in the destination field. However, this time the Switch 8000M's single forwarding database indicates that the 8212zl is on port B1 (VLAN 2), and the 8000M drops the packet instead of forwarding it.
3. Later, the 8212zl switch transmits a packet to the 8000M through the VLAN 1 link, and the 8000M updates its address table to indicate that the 8212zl switch is on port A1 (VLAN 1) instead of port B1 (VLAN 2). Thus, the 8000M's information on the location of the 8212zl switch changes over

time. For this reason, the 8000M discards some packets directed through it for the 8212zl switch, resulting in poor performance and the appearance of an intermittent or broken link.

The Solution. To avoid the preceding problem, use only one cable or port trunk between the single-forwarding and multiple-forwarding database devices, and configure the link with multiple, tagged VLANs.

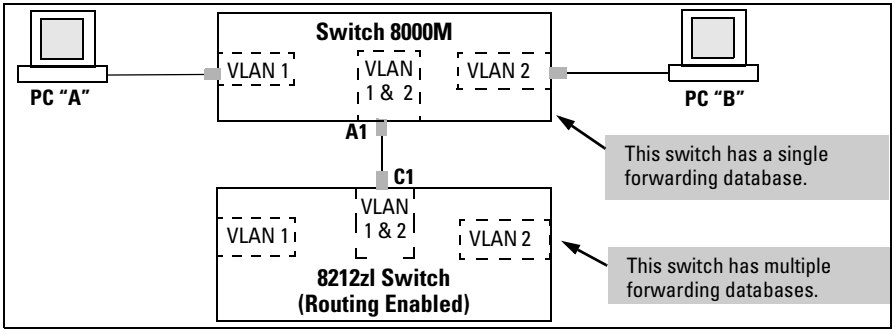


Figure 2-11. Example of a Solution for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

Now, the 8000M forwarding database always lists the 8212zl MAC address on port A1, and the 8000M will send traffic to either VLAN on the 8212zl.

To increase the network bandwidth of the connection between the devices, you can use a trunk of multiple physical links rather than a single physical link.

Multiple Forwarding Database Operation

If you want to connect one of the switches covered by this guide to another switch that has a multiple forwarding database, you can use either or both of the following connection options:

- A separate port or port trunk interface for each VLAN. This results in a forwarding database having multiple instances of the same MAC address with different VLAN IDs and port numbers. (See table 2-5.) The fact that the switches covered by this guide use the same MAC address on all VLAN interfaces causes no problems.
- The same port or port trunk interface for multiple (tagged) VLANs. This results in a forwarding database having multiple instances of the same MAC address with different VLAN IDs, but the same port number.

Allowing multiple entries of the same MAC address on different VLANs enables topologies such as the following:

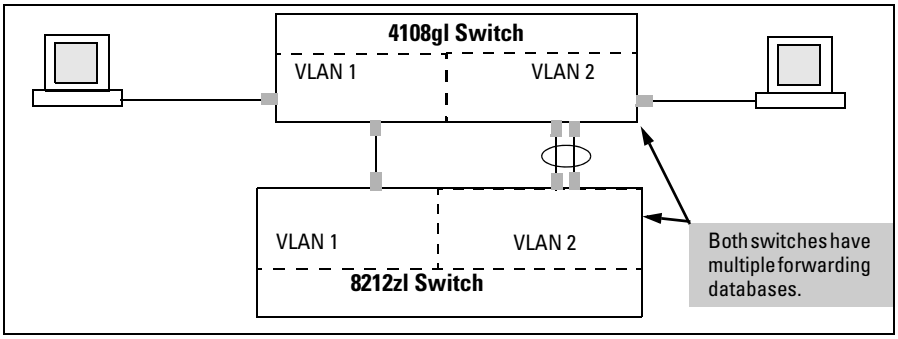


Figure 2-12. Example of a Valid Topology for Devices Having Multiple Forwarding Databases in a Multiple VLAN Environment

Configuring VLANs

Menu: Configuring Port-Based VLAN Parameters

The Menu interface enables you to configure and view port-based VLANs.

Note

The Menu interface configures and displays only port-based VLANs. The CLI configures and displays port-based *and* protocol-based VLANs (page 2-30).

In the factory default state, support is enabled for up to 256 VLANs. (You can reconfigure the switch to support up to 2048 (vids up to 4094) VLANs.) Also, in the default configuration, all ports on the switch belong to the default VLAN and are in the same broadcast/multicast domain. (The default VLAN is also the default Primary VLAN—refer to "The Primary VLAN" on page 2-51.) In addition to the default VLAN, you can configure additional static VLANs by adding new VLAN names and VIDs, and then assigning one or more ports to each VLAN. (The maximum of 2048 VLANs includes the default VLAN, all additional static VLANs you configure, and any dynamic VLANs the switch creates if you enable GVRP—page 3-1.) Note that each port can be assigned to multiple VLANs by using VLAN tagging. (See "802.1Q VLAN Tagging" on page 2-46.)

To Change VLAN Support Settings

This section describes:

- Changing the maximum number of VLANs to support
- Changing the Primary VLAN selection (See "Changing the Primary VLAN" on page 2-39.)
- Enabling or disabling dynamic VLANs (Refer to chapter 3, "GVRP" .)

1. From the Main Menu select:

   2. Switch Configuration

   8. VLAN Menu ...

   1. VLAN Support

   You will then see the following screen:

```
============================ CONSOLE - MANAGER MODE ============================
                    Switch Configuration - VLAN - VLAN Support

  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : No

 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 2-13. The Default VLAN Support Screen

2. Press **[E]** (for **Edit**), then do one or more of the following:
   - To change the maximum number of VLANs, type the new number (1 - 2048 allowed; default 256).
   - To designate a different VLAN as the Primary VLAN, select the **Primary VLAN** field and use the space bar to select from the existing options. (Note that the Primary VLAN must be a static, port-based VLAN.)
   - To enable or disable dynamic VLANs, select the **GVRP Enabled** field and use the Space bar to toggle between options. (For GVRP information, refer to chapter 3, "GVRP" .)

Note

For optimal switch memory utilization, set the number of VLANs at the number you will likely be using or a few more. If you need more VLANs later, you can increase this number, but a switch reboot will be required at that time.

3. Press **[Enter]** and then **[S]** to save the VLAN support configuration and return to the VLAN Menu screen.

If you changed the value for **Maximum VLANs to support**, you will see an asterisk next to the **VLAN Support** option (see below).

An asterisk indicates you must reboot the switch to implement the new Maximum VLANs setting.

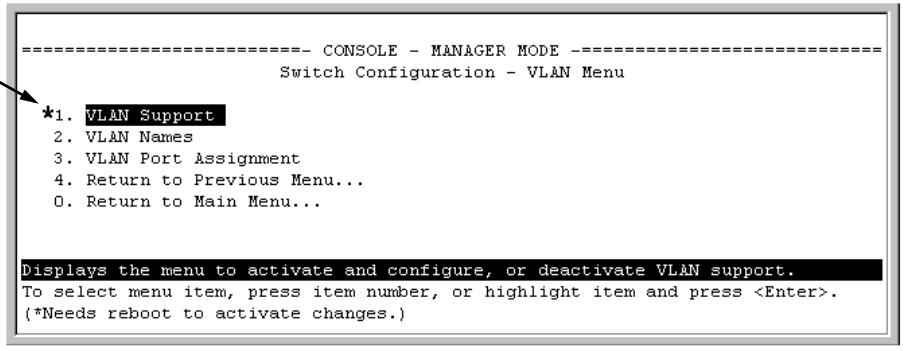

========== CONSOLE - MANAGER MODE ==========
Switch Configuration - VLAN Menu
*1. VLAN Support
2. VLAN Names
3. VLAN Port Assignment
4. Return to Previous Menu...
0. Return to Main Menu...
Displays the menu to activate and configure, or deactivate VLAN support.
To select menu item, press item number, or highlight item and press <Enter>.
(*Needs reboot to activate changes.)

Figure 2-14. VLAN Menu Screen Indicating the Need To Reboot the Switch

- If you changed the VLAN Support option, you must reboot the switch before the Maximum VLANs change can take effect. You can go on to configure other VLAN parameters first, but remember to reboot the switch when you are finished.
- If you did not change the VLAN Support option, a reboot is not necessary.

4. Press **[0]** to return to the Main Menu.

Adding or Editing VLAN Names

Use this procedure to add a new VLAN or to edit the name of an existing VLAN.

1. From the Main Menu select:

   2. Switch Configuration
   8. VLAN Menu …
   2. VLAN Names

   If multiple VLANs are not yet configured you will see a screen similar to figure 2-15:

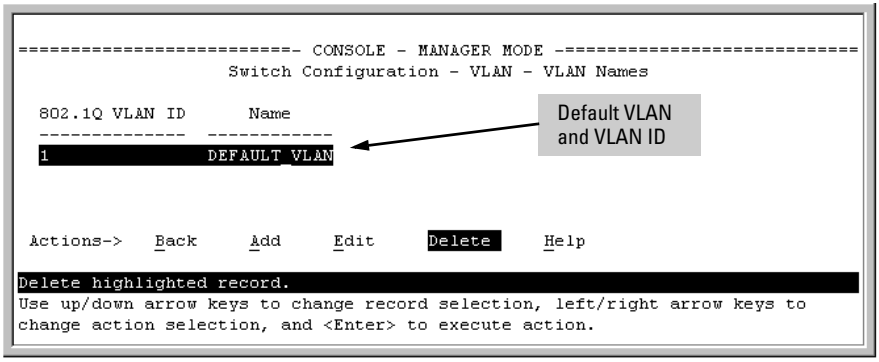

========== CONSOLE - MANAGER MODE ==========
Switch Configuration - VLAN - VLAN Names
802.1Q VLAN ID
Name
1 DEFAULT_VLAN
Default VLAN and VLAN ID
Actions-> Back Add Edit Delete Help
Delete highlighted record.
Use up/down arrow keys to change record selection, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 2-15. The Default VLAN Names Screen

2. Press **[A]** (for **Add**). You will then be prompted for a new VLAN name and VLAN ID:

   802.1Q VLAN ID : 1
   Name : _

3. Type in a VID (VLAN ID number). This can be any number from 2 to 4094 that is not already being used by another VLAN. (The switch reserves "1" for the default VLAN.)

   Remember that a VLAN *must* have the same VID in every switch in which you configure that same VLAN. (GVRP dynamically extends VLANs with correct VID numbering to other switches. Refer to chapter 3, "GVRP" .)

4. Press ↓ to move the cursor to the **Name** line and type the VLAN name (up to 12 characters, with no spaces) of a new VLAN that you want to add, then press **[Enter]**.
   (Avoid these characters in VLAN names: **@, #, $, ^, &, ***, **(**, and **)**.)

5. Press **[S]** (for **Save**). You will then see the VLAN Names screen with the new VLAN listed.

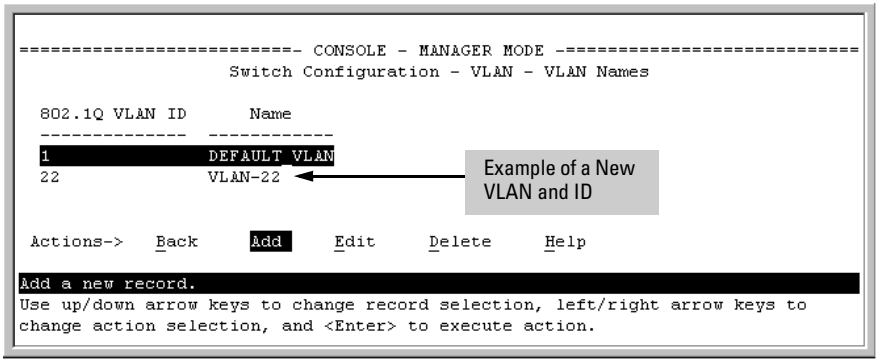

========== CONSOLE - MANAGER MODE ==========
Switch Configuration - VLAN - VLAN Names
802.1Q VLAN ID
Name
1
DEFAULT_VLAN
22
VLAN-22
Example of a New VLAN and ID
Actions-> Back Add Edit Delete Help
Add a new record.
Use up/down arrow keys to change record selection, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 2-16. Example of VLAN Names Screen with a New VLAN Added

6. Repeat steps 2 through 5 to add more VLANs.

   Remember that you can add VLANs until you reach the number specified in the **Maximum VLANs to support** field on the VLAN Support screen (see figure 2-13 on page 2-25). This includes any VLANs added dynamically due to GVRP operation.

7. Return to the VLAN Menu to assign ports to the new VLAN(s) as described in the next section, "Adding or Changing a VLAN Port Assignment".

Adding or Changing a VLAN Port Assignment

Use this procedure to add ports to a VLAN or to change the VLAN assignment(s) for any port. (Ports not specifically assigned to a VLAN are automatically in the default VLAN.)

1. From the Main Menu select:

   2. Switch Configuration

   8. VLAN Menu ...

   3. VLAN Port Assignment

   You will then see a VLAN Port Assignment screen similar to the following:

Note

The "VLAN Port Assignment" screen displays up to 32 static, port-based VLANs in ascending order, by VID. If the switch configuration includes more than 32 such VLANs, use the CLI **show vlans [VID | ports < *port-list* >]** command to list data on VLANs having VIDs numbered sequentially higher than the first 32.

Default: In this example, the "VLAN-22" has been defined, but no ports have yet been assigned to it. ("No" means the port is not assigned to that VLAN.)
Using GVRP? If you plan on using GVRP, any ports you don't want to join should be changed to "Forbid".

A port can be assigned to several VLANs, but only one of those assignments can be "Untagged".

```
=========================- CONSOLE - MANAGER MODE -=============================
                Switch Configuration - VLAN - VLAN Port Assignment

  Port   DEFAULT_VLAN    VLAN-22      |  Port   DEFAULT_VLAN    VLAN-22
  ---- + ------------  ------------   |  ---- + ------------  ------------
 A1    | Untagged      No             |  A8   | Untagged      No
 A2    | Tagged        No             |  A9   | Untagged      No
 A3    | Untagged      No             |  A10  | Untagged      No
 A4    | Untagged      No             |  A11  | Untagged      No
 A5    | Untagged      No             |  A12  | Untagged      No
 A6    | Untagged      No             |  A13  | Untagged      No
 A7    | Untagged      No             |  A14  | Untagged      No

 Actions->   Cancel    Edit    Save    Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 2-17. Example of the Port-Based VLAN Port Assignment Screen in the Menu Interface

2. To change a port's VLAN assignment(s):
   a. Press **[E]** (for **Edit**).
   b. Use the arrow keys to select a VLAN assignment you want to change.
   c. Press the Space bar to make your assignment selection (**No**, **Tagged**, **Untagged**, or **Forbid**).

Note

For GVRP Operation: If you enable GVRP on the switch, "**No**" converts to "**Auto**", which allows the VLAN to dynamically join an advertised VLAN that has the same VID. See "Per-Port Options for Dynamic VLAN Advertising and Joining" on page 3-9.

Untagged VLANs: Only one untagged VLAN is allowed per port. Also, there must be at least one VLAN assigned to each port. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN).

For example, if you want ports A4 and A5 to belong to both DEFAULT_VLAN and VLAN-22, and ports A6 and A7 to belong only to VLAN-22, you would use the settings in figure page 2-30. (This example assumes the default GVRP setting—disabled—and that you do not plan to enable GVRP later.)

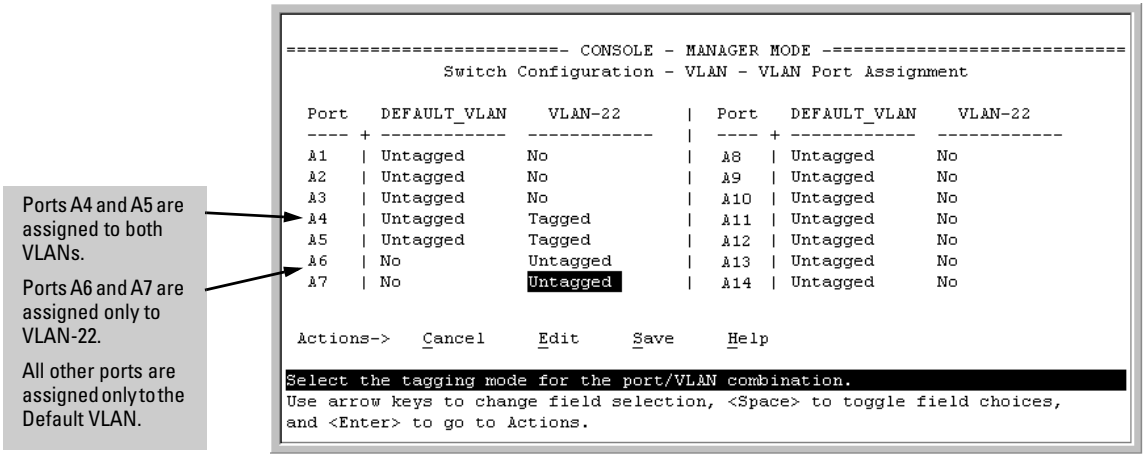

Ports A4 and A5 are assigned to both VLANs.
Ports A6 and A7 are assigned only to VLAN-22.
All other ports are assigned only to the Default VLAN.
========== CONSOLE - MANAGER MODE ==========
Switch Configuration - VLAN - VLAN Port Assignment
Port DEFAULT_VLAN VLAN-22 | Port DEFAULT_VLAN VLAN-22
A1 | Untagged No | A8 | Untagged No
A2 | Untagged No | A9 | Untagged No
A3 | Untagged No | A10 | Untagged No
A4 | Untagged Tagged | A11 | Untagged No
A5 | Untagged Tagged | A12 | Untagged No
A6 | No Untagged | A13 | Untagged No
A7 | No Untagged | A14 | Untagged No
Actions-> Cancel Edit Save Help
Select the tagging mode for the port/VLAN combination.
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.

Figure 2-18. Example of Port-Based VLAN Assignments for Specific Ports

For information on VLAN tags ("Untagged" and "Tagged"), refer to "802.1Q VLAN Tagging" on page 2-46.

d. If you are finished assigning ports to VLANs, press **[Enter]** and then **[S]** (for **Save**) to activate the changes you've made and to return to the Configuration menu. (The console then returns to the VLAN menu.)

3. Return to the Main menu.

CLI: Configuring Port-Based and Protocol-Based VLAN Parameters

In the factory default state, all ports on the switch belong to the (port-based) default VLAN (DEFAULT_VLAN; VID = 1) and are in the same broadcast/multicast domain. (The default VLAN is also the Primary VLAN. For more on this topic, refer to "The Primary VLAN" on page 2-51.) You can configure up to 255 additional static VLANs by adding new VLAN names, and then assigning one or more ports to each VLAN. (The switch accepts a maximum of 2048 (vids numbered up to 4094) VLANs, including the default VLAN and any dynamic VLANs the switch creates if you enable GVRP. Refer to chapter 3, "GVRP" .) Note that each port can be assigned to multiple VLANs by using VLAN tagging. (See "802.1Q VLAN Tagging" on page 2-46.)

VLAN Commands	Page
show vlans	below
show vlans < *vid* >	2-35
show vlans ports <*port-list*>	
max-vlans <1-2048>	2-39
primary-vlan < *vid* >	2-39
[no] vlan < *vid* >	2-41
auto < *port-list* >	2-43 (Available if GVRP enabled.)
forbid	2-43
name < *vlan-name* >	2-43
protocol < *protocol-list* >	2-41
tagged < *port-list* >	2-43
untagged < *port-list* >	2-43
voice	2-60
static-vlan < *vlan-id* >	2-43 (Available if GVRP enabled.)

Displaying the Switch's VLAN Configuration. The **show vlans** command lists the VLANs currently running in the switch, with VID, VLAN name, and VLAN status. Dynamic VLANs appear only if the switch is running with GVRP enabled and one or more ports has dynamically joined an advertised VLAN. (In the default configuration, GVRP is disabled. (Refer to chapter 3, "GVRP" .)

Syntax: show vlans

Maximum VLANs to support: *Shows the number of VLANs the switch can currently support. (Default: 256 Maximum: 2048)*

Primary VLAN: *Refer to "The Primary VLAN" on page 2-51.*

Management VLAN: *Refer to "The Secure Management VLAN" on page 2-52.*

802.1Q VLAN ID: *The VLAN identification number, or VID. Refer to "Terminology" on page 2-7.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "**x**" matches the applicable VID.*

Status:

Port-Based*: Port-Based, static VLAN*

Protocol: *Protocol-Based, static VLAN*

Dynamic: *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3, "GVRP" .)*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-60.*

Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

For example:

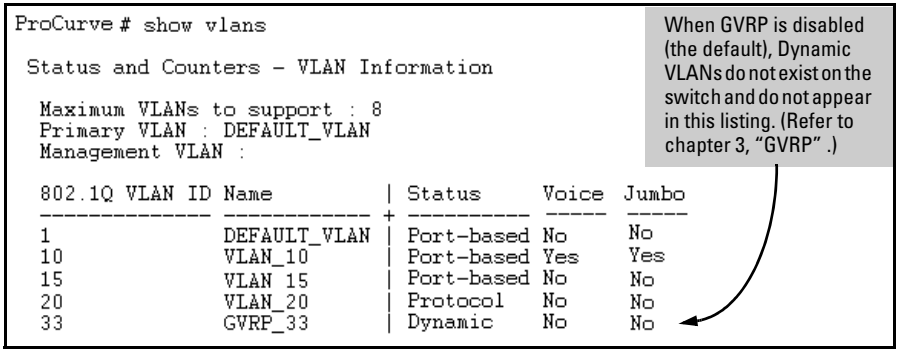

ProCurve # show vlans
Status and Counters - VLAN Information
Maximum VLANs to support : 8
Primary VLAN : DEFAULT_VLAN
Management VLAN :
802.1Q VLAN ID Name | Status Voice Jumbo
1 DEFAULT_VLAN | Port-based No No
10 VLAN_10 | Port-based Yes Yes
15 VLAN 15 | Port-based No No
20 VLAN_20 | Protocol No No
33 GVRP_33 | Dynamic No No
When GVRP is disabled (the default), Dynamic VLANs do not exist on the switch and do not appear in this listing. (Refer to chapter 3, "GVRP" .)

Figure 2-19. Example of "Show VLAN" Listing (GVRP Enabled)

Displaying the VLAN Membership of One or More Ports.

This command shows to which VLAN a port belongs.

Syntax: show vlan ports < *port-list* > [detail]

Displays VLAN information for an individual port or a group of ports, either cumulatively or on a detailed per-port basis.

port-list: *Specify a single port number, a range of ports (for example,* **a1-a16***), or* **all**.

detail: *Displays detailed VLAN membership information on a per-port basis.*

Descriptions of items displayed by the command are provided below.

Port name: *The user-specified port name, if one has been assigned.*

VLAN ID: *The VLAN identification number, or VID.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "**x**" matches the applicable VID.*

Status:

Port-Based*: Port-Based, static VLAN*

Protocol: *Protocol-Based, static VLAN*

Dynamic: *Port-Based, temporary VLAN learned through GVRP.*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN.*

Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

Mode: *Indicates whether a VLAN is tagged or untagged.*

Figure 2-20 is an example of the output when the **detail** option is not used.

```
ProCurve(config)# show vlan ports a1-a24

 Status and Counters - VLAN Information - for ports A1-A24

  VLAN ID Name                | Status     Voice Jumbo
  ------- ------------------- + ---------- ----- -----
  1       DEFAULT_VLAN        | Port-based No    No
  10      VLAN_10             | Port-based Yes   No
  15      VLAN_15             | Protocol   No    No
```

Figure 2-20. Example of "Show VLAN Ports" Cumulative Listing

Figure 2-21 is an example of the output when the **detail** option is used.

```
ProCurve (config)# show vlan ports a1-a3 detail

 Status and Counters - VLAN Information - for ports A1

  VLAN ID Name                 | Status     Voice Jumbo Mode
  ------- -------------------- + ---------- ----- ----- --------
  1       DEFAULT_VLAN         | Port-based No    No    Untagged
  10      VLAN_10              | Port-based Yes   No    Tagged

 Status and Counters - VLAN Information - for ports A2

  VLAN ID Name                 | Status     Voice Jumbo Mode
  ------- -------------------- + ---------- ----- ----- --------
  1       DEFAULT_VLAN         | Port-based No    No    Untagged
  20      VLAN_20              | Protocol   No    No    Untagged

 Status and Counters - VLAN Information - for ports A3

  VLAN ID Name                 | Status     Voice Jumbo Mode
  ------- -------------------- + ---------- ----- ----- --------
  1       DEFAULT_VLAN         | Port-based No    No    Untagged
  33      VLAN_33              | Port-based No    No    Tagged
```

Figure 2-21. Example of "Show VLAN Ports" Detail Listing

Displaying the Configuration for a Particular VLAN . This command uses the VID to identify and display the data for a specific static or dynamic VLAN.

Syntax: show vlans < *vlan-id* >

802.1Q VLAN ID: *The VLAN identification number, or VID. Refer to "Terminology" on page 2-7.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "**x**" matches the applicable VID.*

Status:

Port-Based*: Port-Based, static VLAN*

Protocol: *Protocol-Based, static VLAN*

Dynamic: *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3,"GVRP" in this guide.)*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-60.*

Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

Port Information: *Lists the ports configured as members of the VLAN.*

DEFAULT: *Shows whether a port is a tagged or untagged member of the listed VLAN.*

Unknown VLAN: *Shows whether the port can become a dynamic member of an unknown VLAN for which it receives an advertisement. GVRP must be enabled to allow dynamic joining to occur. Refer to table 3-1 on page 3-8.*

Status: *Shows whether the port is participating in an active link.*

```
ProCurve(config)# show vlans 22
 Status and Counters - VLAN Information - Ports - VLAN 22
  802.1Q VLAN ID : 22
  Name : VLAN22
  Status : Port-based
  Voice : Yes
  Jumbo : No

  Port Information Mode     Unknown VLAN Status
  ---------------- -------- ------------ ----------
  A12              Untagged Learn        Up
  A13              Untagged Learn        Up
  A14              Untagged Learn        Up
  A15              Untagged Learn        Down
  A16              Untagged Learn        Up
  A17              Untagged Learn        Up
  A18              Untagged Learn        Up
```

Figure 2-22. Example of "Show VLAN" for a Specific Static VLAN

Show VLAN lists this data when GVRP is enabled and at least one port on the switch has dynamically joined the designated VLAN.

```
ProCurve# show vlans 33
 Status and Counters - VLAN Information - Ports - VLAN 33

  802.1Q VLAN ID : 33
  Name : GVRP_33
  Status : Dynamic
  Voice : No
  Jumbo : No
  Port Information DEFAULT  Unknown VLAN Status
  ---------------- -------- ------------ ----------
  A6               Auto     Learn        Up
```

Figure 2-23. Example of "Show VLAN" for a Specific Dynamic VLAN

Customizing the Show VLANs Output

The **show vlans custom** command allows you to customize the information displayed when executing the **show vlans** command.

Syntax: show vlans custom [port <*port-list*>] column-list

Select the information that you want to display in the order you want to display it for the **show vlans** *command. You can display information for one port or range of ports. If <port-list> isn't specified, then all ports display.*

Fields that can be included in the customized display are shown in the table below.

Field	Display	Example	Default
id	VLAN Id	5	6
name	VLAN Name	Vlan55	32
status	Status	Port-based	10
voice	Voice enabled	No	5
jumbo	Jumbos enabled	No	5
ipconfig	How the ip address was configured	Manual Disabled DHCP/BootP	10
ipaddr (IPv4) ipaddr (IPv6)	the IP address(es)	10.10.10.3 fe80::212:79ff:fe8d:8000	15 for IPv4 46 for IPv6
ipmask	The subnet mask(s)	255.255.255.6 /64 (prefix for IPv6 is in format "/XX")	15
proxyarp	Whether proxy arp is configured	No	5
localproxyarp	Whether local proxy arp is configured	No	9
state	"Up" if at least one port is up	Up	5

The example in Figure 2-24 displays **id** at its default width, and will show up to 20 characters of the VLAN **name**. The columns selected for display are separated by spaces.

```
ProCurve(config)# show vlan custom A1-A3 id name:20 ipaddr state

Status and Counters - VLAN Information - Custom view

 VLANID VLAN name             IP Addr                          State
 ------ --------------------  -------------------------------- -----
 1      DEFAULT_VLAN          15.255.134.74                    Up
 33     Vlan33                10.10.10.01                      Up
 44     Vlan44                15.255.164.13                    Up
 55     Vlan55                15.255.178.2                     Down
                              15.255.178.3
                              15.255.178.4
 60     Vlan60                fe80::212:79ff:fe8d:8000%vlan60  Up
```

Figure 2-24. Example of show vlan custom Command

If the width of the column requested is smaller than the header name of the column, the display of the header name is truncated.

```
ProCurve(config)# show vlan custom id
Status and Counters - VLAN Information - Custom view

 VLANID
 ------
 1
 33
 44

ProCurve(config)# show vlan custom id:2
Status and Counters - VLAN Information - Custom view

 VL
 --
 1
 33
 44
```

Figure 2-25. Example of Column Headers

The total output will wrap if it is longer than the terminal width (for example, 80 characters). It is not truncated.

Creating an Alias for Show VLAN Commands

You can create an alias for a frequently used **show vlans custom** command to avoid entering the selected columns each time you use the command.

```
ProCurve(config)# alias showvlanstatus = “show vlan custom A1-A3 id name:20
status”

ProCurve(config)# showvlanstatus
Status and Counters - VLAN Information - Custom view

 VLANID VLAN name            Status
 ------ -------------------- ----------
 1      DEFAULT_VLAN         Port-based
 33     Vlan33               Port-based
```

Figure 2-26. Example of the alias Command

Note on Using Pattern Matching with the "Show VLANs Custom" Command

If you have included a pattern matching command to search for a field in the output of the **show vlan custom** command and the **show vlans custom** command produces an error, the error message may not be visible and the output is empty. For example, if you enter a command that produces an error (vlan is misspelled) with the pattern matching **include** option:

```
ProCurve(config)# show vlans custom 1-3 name vlun |
include vlan1
```

the output may be empty. It is advisable to try the **show vlans custom** command first to ensure there is output, and then enter the command again with the pattern matching option.

Changing the Number of VLANs Allowed on the Switch

In the default VLAN configuration, the switch allows a maximum of 256 VLANs. You can specify any value from 1 to 2048.

Syntax: max-vlans < 1-2048 >

Specifies the maximum number of VLANs to allow. (If GVRP is enabled, this setting includes any dynamic VLANs on the switch.) As part of implementing a new setting, you must execute a **write memory** *command (to save the new value to the startup-config file) and then reboot the switch.*

Note: *If multiple VLANs exist on the switch, you cannot reset the maximum number of VLANs to a value smaller than the current number of VLANs.*

For example, to reconfigure the switch to allow 10 VLANs:

Note that you can execute these three steps at another time.

```
ProCurve(config)# max-vlans 10
Command will take effect after saving configuration and reboot.
ProCurve(config)# write memory
ProCurve(config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 2-27. Example of Command Sequence for Changing the Number of VLANs

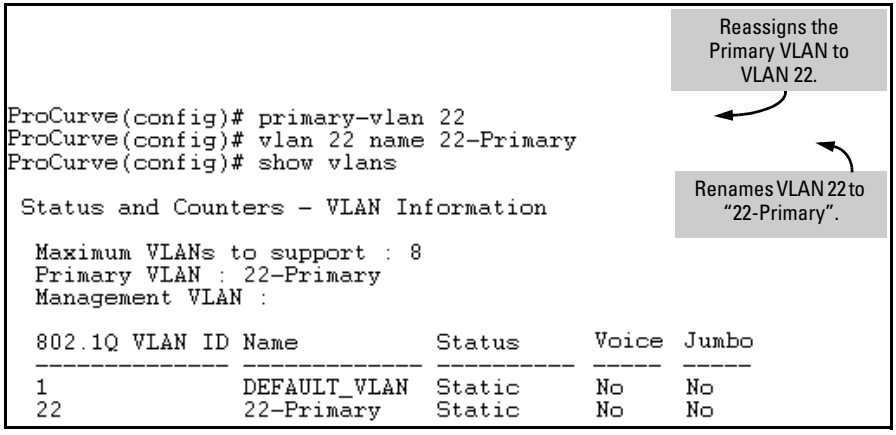
Changing the Primary VLAN. In the default VLAN configuration, the port-based default VLAN (**DEFAULT_VLAN**) is the Primary VLAN. However, you can reassign the Primary VLAN to any port-based, static VLAN on the switch. (For

more on the Primary VLAN, refer to "The Primary VLAN" on page 2-51.) To identify the current Primary VLAN and list the available VLANs and their respective VIDs, use **show vlans**.

Syntax: primary-vlan < *vid* | *ascii-name-string* >

> *Reassigns the Primary VLAN function. Re-assignment must be to an existing, port-based, static VLAN. (The switch will not reassign the Primary VLAN function to a protocol VLAN.) If you re-assign the Primary VLAN to a non-default VLAN, you cannot later delete that VLAN from the switch until you again re-assign the Primary VLAN to another port-based, static VLAN.*

For example, if you wanted to reassign the Primary VLAN to VLAN 22 and rename the VLAN with "22-Primary" and display the result:




```
ProCurve(config)# primary-vlan 22
ProCurve(config)# vlan 22 name 22-Primary
ProCurve(config)# show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 8
  Primary VLAN : 22-Primary
  Management VLAN :

  802.1Q VLAN ID Name          Status      Voice  Jumbo
  -------------- ------------- ----------- -----  -----
  1              DEFAULT_VLAN  Static      No     No
  22             22-Primary    Static      No     No
```

Figure 2-28. Example of Reassigning Primary VLAN and Changing the VLAN Name

Creating a New Static VLAN (Port-Based or Protocol-Based)
Changing the VLAN Context Level. The **vlan < *vid*>** command operates in the global configuration context to either configure a static VLAN and/or take the CLI to the specified VLAN's context.

Syntax: vlan < *vid* | *ascii-name-string* >
[no] vlan < *vid* >

If < vid > does not exist in the switch, this command creates a port-based VLAN with the specified < vid >. If the command does not include options, the CLI moves to the newly created VLAN context. If you do not specify an optional name, the switch assigns a name in the default format: **VLAN*n*** *where* ***n*** *is the < vid > assigned to the VLAN. If the VLAN already exists and you enter either the* ***vid*** *or the* ***ascii-name-string****, the CLI moves to the specified VLAN's context.*

The **[no]** *form of the command deletes the VLAN as follows:*

- *If one or more ports belong only to the VLAN to be deleted, the CLI notifies you that these ports will be moved to the default VLAN and prompts you to continue the deletion. For member ports that also belong to another VLAN, there is no "move" prompt.*

[protocol < ipx | ipv4 | ipv6 | arp | appletalk | sna | netbeui >]

Configures a static, protocol VLAN of the specified type. If multiple protocols are configured in the VLAN, then the **[no]** *form removes the specified protocol from the VLAN. If a protocol VLAN is configured with only one protocol type and you use the* **[no]** *form of this command to remove that protocol, the switch changes the protocol VLAN to a port-based VLAN if the VLAN does not have an untagged member port. (If an untagged member port exists on the protocol VLAN, you must either convert the port to a tagged member or remove the port from the VLAN before removing the last protocol type from the VLAN.)*

Note: *If you create an IPv4 protocol VLAN, you must also assign the ARP protocol option to the VLAN to provide IP address resolution. Otherwise, IP packets are not deliverable. A "Caution" message appears in the CLI if you configure IPv4 in protocol VLAN that does not already include the arp protocol option. The same message appears if you add or delete another protocol in the same VLAN.*

name < *ascii-name-string* >

When included in a **vlan** *command for creating a new static VLAN, specifies a non-default VLAN name. Also used to change the current name of an existing VLAN. (Avoid spaces and the following characters in the <* ***ascii-name-string*** *> entry:* **@**, **#**, **$**, **^**, **&**, *****, **(**, and **)**. *To include a blank space in a VLAN name, enclose the name in single or double quotes ('...' or "...").*

[voice]

Designates a VLAN for VoIP use. For more on this topic, refer to "Voice VLANs" on page 2-60.

For example, to create a new, port-based, static VLAN with a VID of 100:

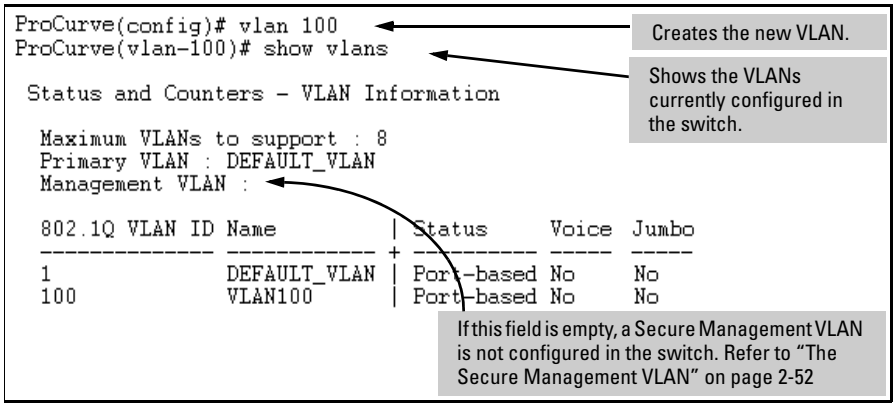

ProCurve(config)# vlan 100
ProCurve(vlan-100)# show vlans
Status and Counters - VLAN Information
Maximum VLANs to support : 8
Primary VLAN : DEFAULT_VLAN
Management VLAN :
802.1Q VLAN ID Name | Status Voice Jumbo
1 DEFAULT_VLAN | Port-based No No
100 VLAN100 | Port-based No No
Creates the new VLAN.
Shows the VLANs currently configured in the switch.
If this field is empty, a Secure Management VLAN is not configured in the switch. Refer to "The Secure Management VLAN" on page 2-52

Figure 2-29. Example of Creating a New, Port-Based, Static VLAN

To go to a different VLAN context level, such as to the default VLAN:

```
ProCurve(vlan-100)# vlan default_vlan
ProCurve(vlan-1) _
```

Deleting a VLAN . If ports B1-B5 belong to both VLAN 2 and VLAN 3, and ports B6-B10 belong to VLAN 3 only, then deleting VLAN 3 causes the CLI to prompt you to approve moving ports B6 - B10 to VLAN 1 (the default VLAN). (Ports B1-B5 are not moved because they still belong to another VLAN.)

```
ProCurve(config)# no vlan 3
The following ports will be moved to the default VLAN:
B6-B10
Do you want to continue? [y/n] y
ProCurve(config)#
```

Converting a Dynamic VLAN to a Static VLAN. Use this feature if you want to convert a dynamic, port-based VLAN membership to a static, port-based VLAN membership. This is necessary if you want to make the VLAN permanent on the switch.

Syntax: static-vlan < *vlan-id* >

Converts a dynamic, port-based VLAN membership to a static, port-based VLAN membership. (Allows port-based VLANs only). For this command, < *vlan-id* > *refers to the VID of the dynamic VLAN membership. (Use* **show vlan** to help identify the VID you need to use.) *This command requires that GVRP is running on the switch and a port is currently a dynamic member of the selected VLAN. After you convert a dynamic VLAN to static, you must configure the switch's per-port participation in the VLAN in the same way that you would for any static VLAN. (For GVRP and dynamic VLAN operation, refer to chapter 3, "GVRP" .)*

For example, suppose a dynamic VLAN with a VID of 125 exists on the switch. The following command converts the VLAN to a port-based, static VLAN.

```
ProCurve(config)# static-vlan 125
```

Configuring Static VLAN Per-Port Settings. The **vlan <***vlan-id***>** command, used with the options listed below, changes the name of an existing static VLAN and changes the per-port VLAN membership settings.

Note

You can use these options from the configuration level by beginning the command with **vlan <** *vid* **>**, or from the context level of the specific VLAN by just typing the command option.

Syntax: [no] vlan < *vid* >

tagged < *port-list* >

Configures the indicated port(s) as **Tagged** *for the specified VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**.

untagged < *port-list* >

Configures the indicated port(s) as **Untagged** *for the specified VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**.

forbid < *port-list* >

Used in port-based VLANs to configures **<** *port-list* **>** *as "forbidden" to become a member of the specified VLAN, as well as other actions. Does not operate with protocol VLANs. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**. *Refer to chapter 3, "GVRP", in this guide.*

auto < *port-list* >

Available if GVRP is enabled on the switch. Returns the per-port settings for the specified VLAN to **Auto** *operation. Note that* **Auto** *is the default per-port setting for a static VLAN if GVRP is running on the switch.* (For information on dynamic VLAN and GVRP operation, refer to *chapter 3, "GVRP", in this guide*.)

For example, suppose you have a VLAN named VLAN100 with a VID of 100, and all ports are set to **No** for this VLAN. To change the VLAN name to "**Blue_Team**" and set ports A1 - A5 to **Tagged**, you would use these commands:

```
ProCurve(config)# vlan 100 name Blue_Team
ProCurve(config)# vlan 100 tagged a1-a5
```

To move to the vlan 100 context level and execute the same commands:

```
ProCurve(config)# vlan 100
ProCurve(vlan-100)# name Blue_Team
ProCurve(vlan-100)# tagged a1-a5
```

Similarly, to change the tagged ports in the above examples to **No** (or **Auto**, if GVRP is enabled), you could use either of the following commands.

At the global config level, use:

```
ProCurve(config)# no vlan 100 tagged a1-a5
```

- or -

At the VLAN 100 context level, use:

```
ProCurve(vlan-100)# no tagged a1-a5
```

Note

You cannot use these commands with dynamic VLANs. Attempting to do so results in the message "**VLAN already exists.**" and no change occurs.

Web: Viewing and Configuring VLAN Parameters

In the web browser interface you can do the following:

- Add VLANs
- Rename VLANs
- Remove VLANs
- Configure VLAN tagging mode per-port
- Configure GVRP mode
- Select a new Primary VLAN

To configure other static VLAN port parameters, you will need to use either the CLI or the menu interface (available by Telnet from the web browser interface).

1. Click on the Configuration tab.
2. Click on **[Vlan Configuration]**.
3. Click on **[Add/Remove VLANs]**.

For web-based Help on how to use the web browser interface screen, click on the **[?]** button provided on the web browser screen.

802.1Q VLAN Tagging

General Applications:

- The switch requires VLAN tagging on a given port if more than one VLAN of the same type uses the port. When a port belongs to two or more VLANs of the same type, they remain as separate broadcast domains and cannot receive traffic from each other without routing. (If multiple, *non-routable* VLANs exist in the switch—such as NETbeui protocol VLANs— then they cannot receive traffic from each other under any circumstances.)
- The switch requires VLAN tagging on a given port if the port will be receiving inbound, tagged VLAN traffic that should be forwarded. Even if the port belongs to only one VLAN, it forwards inbound tagged traffic only if it is a tagged member of that VLAN.
- If the only authorized, inbound VLAN traffic on a port arrives untagged, then the port must be an untagged member of that VLAN. This is the case where the port is connected to a non 802.1Q-compliant device or is assigned to only one VLAN.

For example, if port 7 on an 802.1Q-compliant switch is assigned to only the Red VLAN, the assignment can remain “untagged” because the port will forward traffic only for the Red VLAN. However, if both the Red and Green VLANs are assigned to port 7, then at least one of those VLAN assignments must be “tagged” so that Red VLAN traffic can be distinguished from Green VLAN traffic. Figure 2-30 shows this concept:

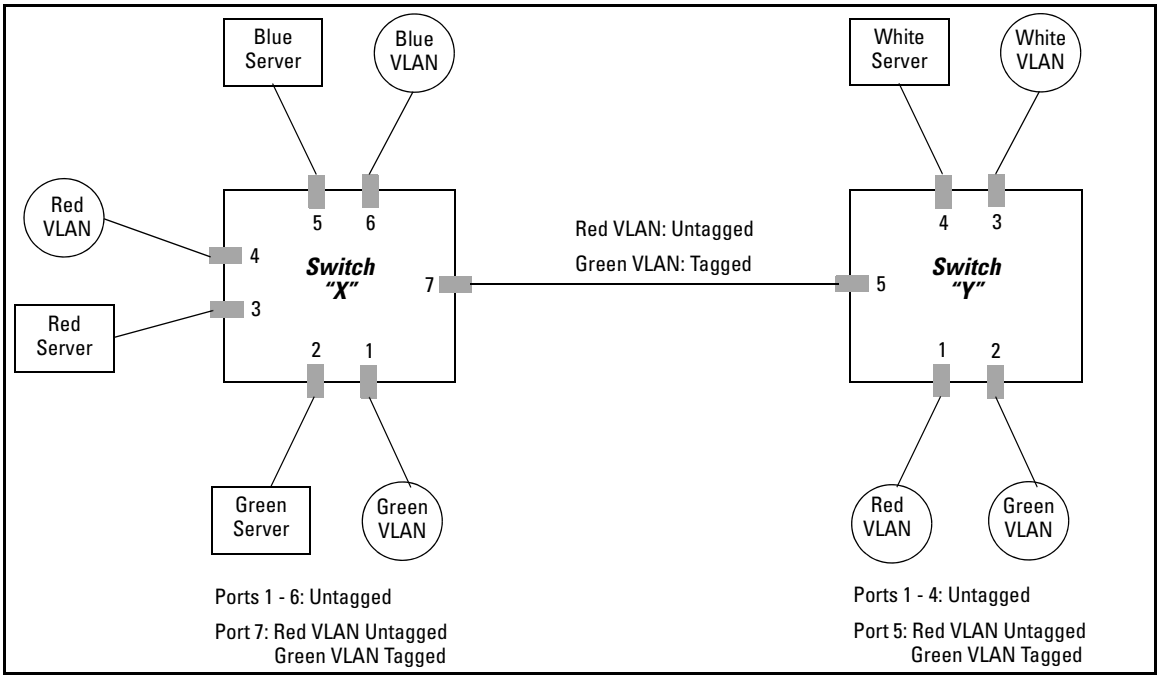


Figure 2-30. Example of Tagged and Untagged VLAN Port Assignments

- In switch X:
  - VLANs assigned to ports X1 - X6 can all be untagged because there is only one VLAN assignment per port. Red VLAN traffic will go out only the Red ports; Green VLAN traffic will go out only the Green ports, and so on. Devices connected to these ports do not have to be 802.1Q-compliant.
  - However, because both the Red VLAN and the Green VLAN are assigned to port X7, at least one of the VLANs must be tagged for this port.
- In switch Y:
  - VLANs assigned to ports Y1 - Y4 can all be untagged because there is only one VLAN assignment per port. Devices connected to these ports do not have to be 802.1Q-compliant.
  - Because both the Red VLAN and the Green VLAN are assigned to port Y5, at least one of the VLANs must be tagged for this port.
- In both switches: The ports on the link between the two switches must be configured the same. As shown in figure 2-30 (above), the Red VLAN must be untagged on port X7 and Y5 and the Green VLAN must be tagged on port X7 and Y5, or vice-versa.

Note

Each 802.1Q-compliant VLAN must have its own unique VID number, and that VLAN *must* be given the same VID in every device in which it is configured. That is, if the Red VLAN has a VID of 10 in switch X, then 10 must also be used for the Red VID in switch Y.

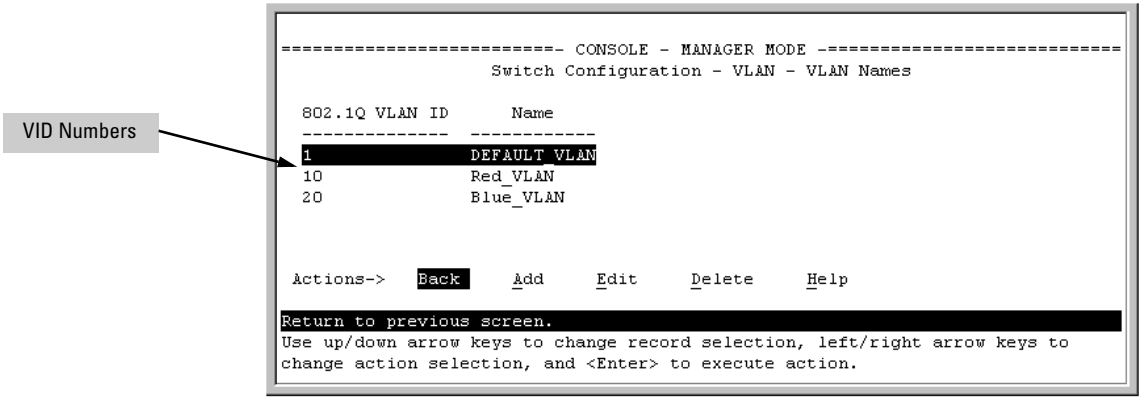

VID Numbers
========== CONSOLE - MANAGER MODE ==========
Switch Configuration - VLAN - VLAN Names
802.1Q VLAN ID
Name
1 DEFAULT_VLAN
10 Red_VLAN
20 Blue_VLAN
Actions-> Back Add Edit Delete Help
Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 2-31. Example of VLAN ID Numbers Assigned in the VLAN Names Screen

VLAN tagging gives you several options:

- Since the purpose of VLAN tagging is to allow multiple VLANs on the same port, any port that has only one VLAN assigned to it can be configured as "Untagged" (the default) if the authorized inbound traffic for that port arrives untagged.
- Any port with two or more VLANs of the same type can have one such VLAN assigned as "Untagged". All other VLANs of the same type must be configured as "Tagged". That is:

Port-Based VLANs	Protocol VLANs
A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing the same type, the port can be an untagged member of only one such VLAN.
A port can be a tagged member of any port-based VLAN. See above.	A port can be a tagged member of any protocol-based VLAN. See above.
Note: A given VLAN ***must*** have the same VID on all 802.1Q-compliant devices in which the VLAN occurs. Also, the ports connecting two 802.1Q devices should have identical VLAN configurations.	

- If all end nodes on a port comply with the 802.1Q standard and are configured to use the correct VID, then, you can configure all VLAN assignments on a port as "Tagged" if doing so either makes it easier to manage your VLAN assignments, or if the authorized, inbound traffic for all VLANs on the port will be tagged.

For a summary and flowcharts of untagged and tagged VLAN operation on inbound traffic, refer to the following under "VLAN Operating Rules" on pages 2-15 through 2-18:

- "Inbound Tagged Packets"
- "Untagged Packet Forwarding" and figure 2-7
- "Tagged Packet Forwarding" and figure 2-8

Example. In the following network, switches X and Y and servers S1, S2, and the AppleTalk server are 802.1Q-compliant. (Server S3 could also be 802.1Q-compliant, but it makes no difference for this example.) This network includes both protocol-based (AppleTalk) VLANs and port-based VLANs.

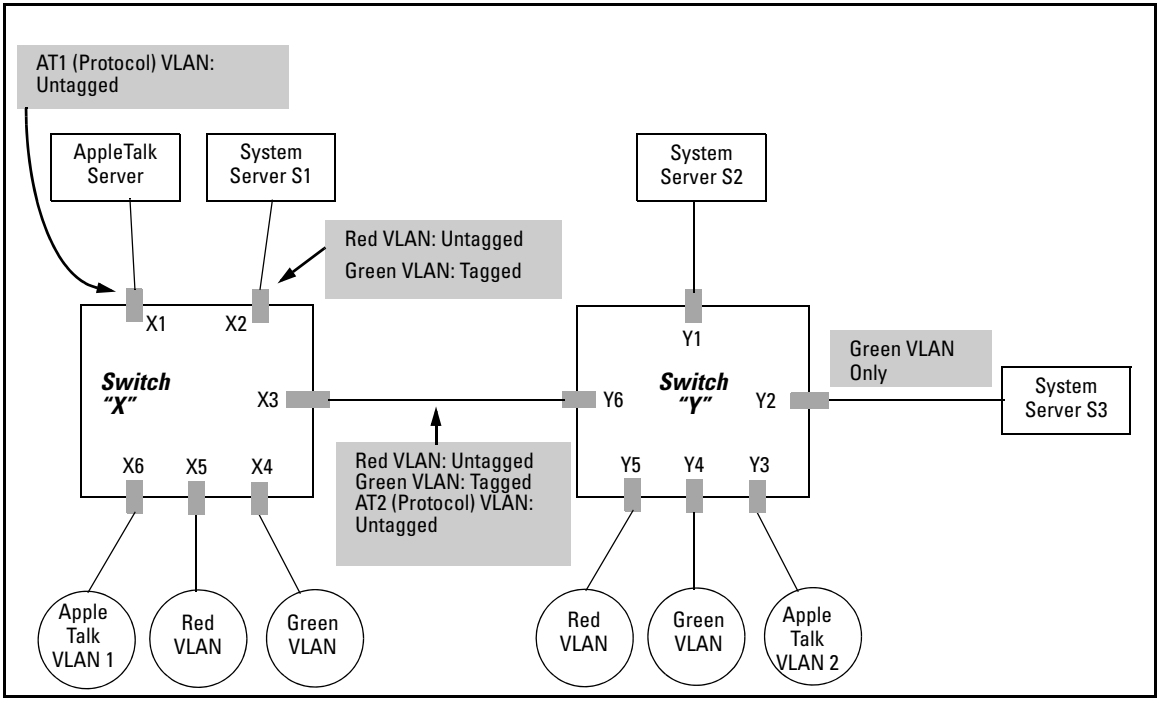


Figure 2-32. Example of Networked 802.1Q-Compliant Devices with Multiple VLANs on Some Ports

- The VLANs assigned to ports X4 - X6, Y2 - Y5 can all be untagged because there is only one VLAN assigned per port.
- Port X1 has two AppleTalk VLANs assigned, which means that one VLAN assigned to this port can be untagged and the other must be tagged.
- Ports X2 and Y1 have two port-based VLANs assigned, so one can be untagged and the other must be tagged on both ports.
- Ports X3 and Y6 have two port-based VLANs and one protocol-based VLAN assigned. Thus, one port-based VLAN assigned to this port can be untagged and the other must be tagged. Also, since these two ports share the same link, their VLAN configurations must match.

Switch X					***Switch Y***				
Port	**AT-1 VLAN**	**AT-2 VLAN**	**Red VLAN**	**Green VLAN**	**Port**	**AT-1 VLAN**	**AT-2 VLAN**	**Red VLAN**	**Green VLAN**
X1	Untagged	Tagged	No*	No*	Y1	No*	No*	Untagged	Tagged
X2	No*	No*	Untagged	Tagged	Y2	No*	No*	No*	Untagged
X3	No*	Untagged	Untagged	Tagged	Y3	No*	Untagged	No*	No*
X4	No*	No*	No*	Untagged	Y4	No*	No*	No*	Untagged
X5	No*	No*	Untagged	No*	Y5	No*	No*	Untagged	No*
X6	Untagged	No*	No*	No*	Y6	No	Untagged	Untagged	Tagged

*"No" means the port is not a member of that VLAN. For example, port X3 is not a member of the Red VLAN and does not carry Red VLAN traffic. Also, if GVRP were enabled (port-based only), "Auto" would appear instead of "No".

Note

VLAN configurations on ports connected by the same link must match. Because ports X2 and Y5 are opposite ends of the same point-to-point connection, both ports must have the same VLAN configuration; that is, both ports configure the Red VLAN as "Untagged" and the Green VLAN as "Tagged".

Special VLAN Types

VLAN Support and the Default VLAN

In the factory default configuration, VLAN support is enabled and all ports on the switch belong to the port-based, default VLAN (named DEFAULT_VLAN). This places all ports in the switch into one physical broadcast domain. In the factory-default state, the default VLAN is also the *Primary* VLAN.

You can partition the switch into multiple virtual broadcast domains by configuring one or more additional VLANs and moving ports from the default VLAN to the new VLANs. (The switch supports up to 2048 (vids numbered up to 4094) static and dynamic VLANs.) You can change the name of the default VLAN, but you cannot change the default VLAN's VID (which is always "1"). Although you can remove all ports from the default VLAN (by placing them in another port-based VLAN), this VLAN is always present; that is, you cannot delete it from the switch.

For details on port VLAN settings, refer to "Configuring Static VLAN Per-Port Settings" on page 2-43

The Primary VLAN

Because certain features and management functions run on only one VLAN in the switch, and because DHCP and Bootp can run per-VLAN, there is a need for a dedicated VLAN to manage these features and ensure that multiple instances of DHCP or Bootp on different VLANs do not result in conflicting configuration values for the switch. The *Primary* VLAN is the VLAN the switch uses to run and manage these features and data. In the factory-default configuration, the switch designates the default VLAN (DEFAULT_VLAN; VID = 1) as the Primary VLAN. However, to provide more control in your network, you can designate another static, port-based VLAN as primary. To summarize, *designating a non-default VLAN as primary* means that:

- The switch reads DHCP responses on the Primary VLAN instead of on the default VLAN. (This includes such DHCP-resolved parameters as the TimeP server address, Default TTL, and IP addressing—including the Gateway IP address—when the switch configuration specifies DHCP as the source for these values.)

- The default VLAN continues to operate as a standard VLAN (except, as noted above, you cannot delete it or change its VID).
- Any ports not specifically assigned to another VLAN will remain assigned to the Default VLAN, regardless of whether it is the Primary VLAN.

Candidates for Primary VLAN include any static, port-based VLAN currently configured on the switch. (Protocol-Based VLANs and dynamic—GVRP-learned—VLANs that have not been converted to a static VLAN cannot be the Primary VLAN.) To display the current Primary VLAN, use the CLI **show vlan** command.

Note

If you configure a non-default VLAN as the Primary VLAN, you cannot delete that VLAN unless you first select a different VLAN to serve as primary.

If you manually configure a gateway on the switch, it ignores any gateway address received via DHCP or Bootp.

To change the Primary VLAN configuration, refer to "Changing the Primary VLAN" on page 2-39.

The Secure Management VLAN

Configuring a secure Management VLAN creates an isolated network for managing the ProCurve switches that support this feature. If you configure a secure Management VLAN, access to the VLAN and to the switch's management functions (Menu, CLI, and web browser interface) is available only through ports configured as members.

- Multiple ports on the switch can belong to the Management VLAN. This allows connections for multiple management stations you want to have access to the Management VLAN, while at the same time allowing Management VLAN links between switches configured for the same Management VLAN.
- Only traffic from the Management VLAN can manage the switch, which means that only the workstations and PCs connected to ports belonging to the Management VLAN can manage and reconfigure the switch.

Figure 2-33 illustrates use of the Management VLAN feature to support management access by a group of management workstations.

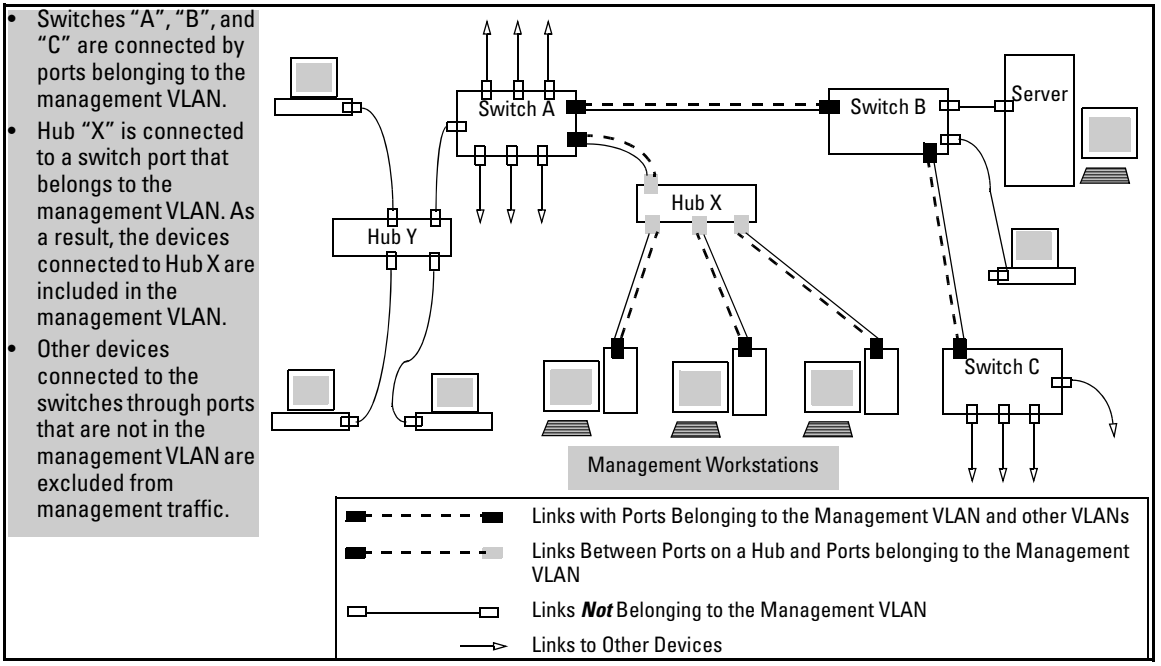

• Switches "A", "B", and "C" are connected by ports belonging to the management VLAN.
• Hub "X" is connected to a switch port that belongs to the management VLAN. As a result, the devices connected to Hub X are included in the management VLAN.
• Other devices connected to the switches through ports that are not in the management VLAN are excluded from management traffic.
Switch A
Switch B
Server
Hub X
Hub Y
Switch C
Management Workstations
Links with Ports Belonging to the Management VLAN and other VLANs
Links Between Ports on a Hub and Ports belonging to the Management VLAN
Links *Not* Belonging to the Management VLAN
Links to Other Devices

Figure 2-33. Example of Potential Security Breaches

In figure 2-34, Workstation 1 has management access to all three switches through the Management VLAN, while the PCs do not. This is because configuring a switch to recognize a Management VLAN automatically excludes attempts to send management traffic from any other VLAN.

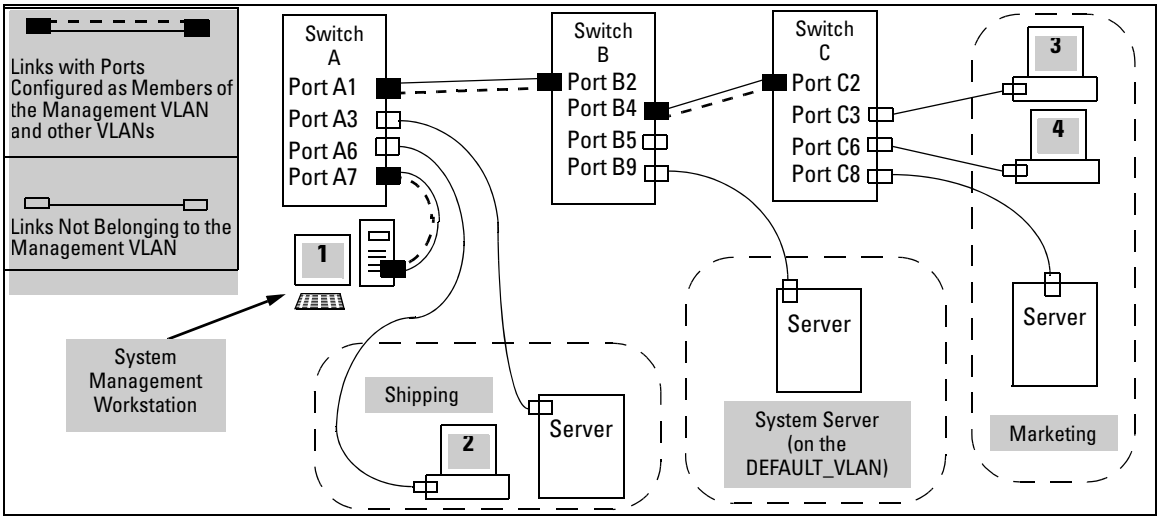


Figure 2-34. Example of Management VLAN Control in a LAN

Table 2-7. VLAN Membership in Figure 2-34

Switch	A1	A3	A6	A7	B2	B4	B5	B9	C2	C3	C6	C8
Management VLAN (VID = 7)	**Y**	N	N	**Y**	**Y**	**Y**	N	N	**Y**	N	N	N
Marketing VLAN (VID = 12)	N	N	N	N	N	N	N	N	N	**Y**	**Y**	**Y**
Shipping Dept. VLAN (VID = 20)	N	**Y**	**Y**	N	N	N	N	N	N	N	N	N
DEFAULT-VLAN (VID = 1)	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**

Preparation

1. Determine a VID and VLAN name suitable for your Management VLAN.
2. Plan your Management VLAN topology to use ProCurve switches that support this feature. (Refer to page 2-52.) The ports belonging to the Management VLAN should be only the following:
   - Ports to which you will connect authorized management stations (such as Port A7 in figure 2-34.)
   - Ports on one switch that you will use to extend the Management VLAN to ports on other ProCurve switches (such as ports A1 and B2 or B4 and C2 in figure 2-34 on page 2-54.).

Hubs dedicated to connecting management stations to the Management VLAN can also be included in the above topology. Note that any device connected to a hub in the Management VLAN will also have Management VLAN access.

3. Configure the Management VLAN on the selected switch ports.
4. Test the management VLAN from all of the management stations authorized to use the Management VLAN, including any SNMP-based network management stations. Ensure that you include testing any Management VLAN links between switches.

Note

If you configure a Management VLAN on a switch by using a Telnet connection through a port that is not in the Management VLAN, then you will lose management contact with the switch if you log off your Telnet connection or execute **write memory** and **reboot** the switch.

Configuration

Syntax: [no] management-vlan < *vlan-id* / *vlan-name* >

Configures an existing VLAN as the management VLAN. The **no** *form disables the management VLAN and returns the switch to its default management operation. Default: Disabled. In this case, the VLAN returns to standard VLAN operation.*

For example, suppose you have already configured a VLAN named **My_VLAN** with a VID of 100. Now you want to configure the switch to do the following:

- Use **My_VLAN** as a Management VLAN (tagged, in this case) to connect port A1 on switch "A" to a management station. (The management station includes a network interface card with 802.1Q tagged VLAN capability.)
- Use port A2 to extend the Management VLAN to port B1 (which is already configured as a tagged member of **My_VLAN**) on an adjacent Procurve switch that supports the Management VLAN feature.

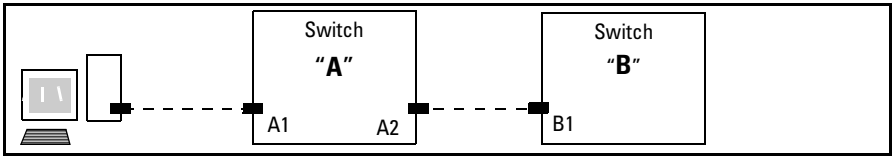


Figure 2-35. Illustration of Configuration Example

```
ProCurve (config)# management-vlan 100
ProCurve (config)# vlan 100 tagged a1
ProCurve (config)# vlan 100 tagged a2
```

Using DHCP to Obtain an IP Address

You can use DHCP to obtain an IPv4 address for your Management VLAN or a client on that VLAN. The following examples illustrate when an IP address will be received from the DHCP server.

1. If Blue_VLAN is configured as the Management VLAN and the DHCP server is also on Blue_VLAN, Blue_VLAN receives an IP address. Because DHCP Relay does not forward onto or off of the Management VLAN, devices on Red_VLAN cannot get an IP address from the DHCP server on Blue_VLAN (Management VLAN) and Red_VLAN does not receive an IP address. See figure 2-36.

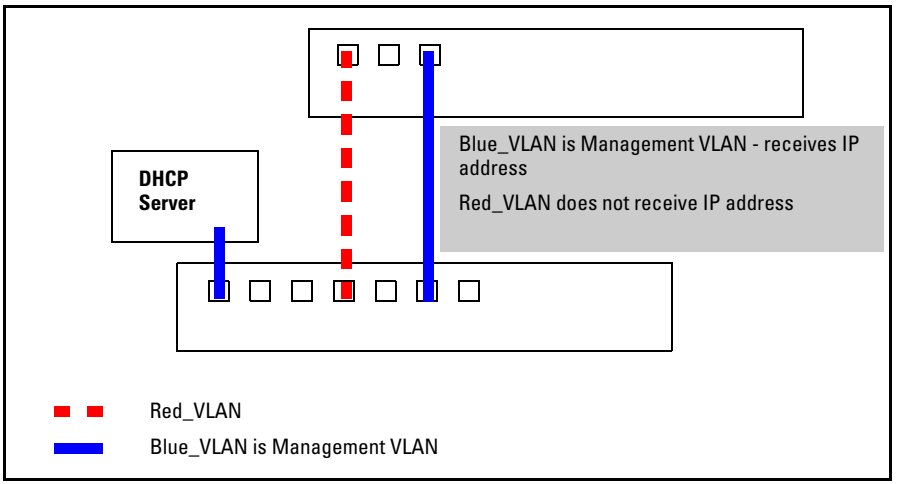


Figure 2-36. Example of DHCP Server on Management VLAN

2. If Red_VLAN is configured as the Management VLAN and the DHCP server is on Blue_VLAN, Blue_VLAN receives an IP address but Red_VLAN does not. See figure 2-37.

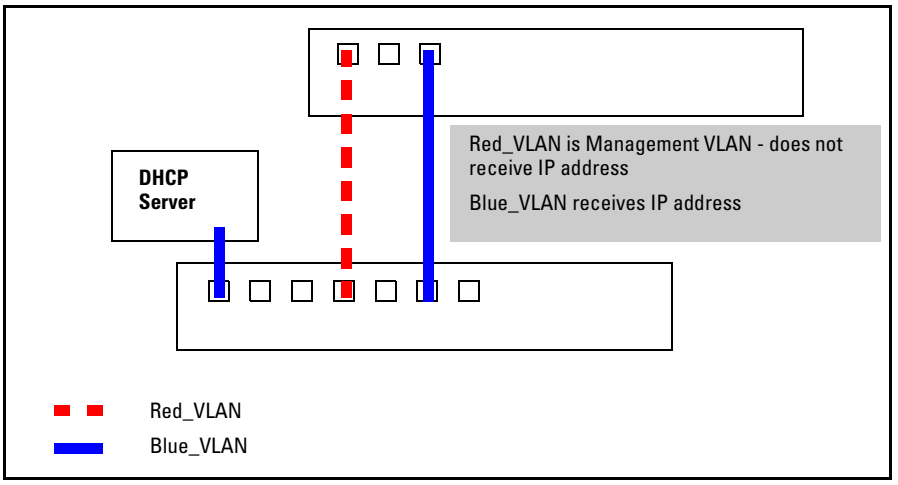


Figure 2-37. Example of DHCP Server on Different VLAN from the Management VLAN

3. If no Management VLAN is configured, both Blue_VLAN and Red_VLAN receive IP addresses. See figure 2-38.

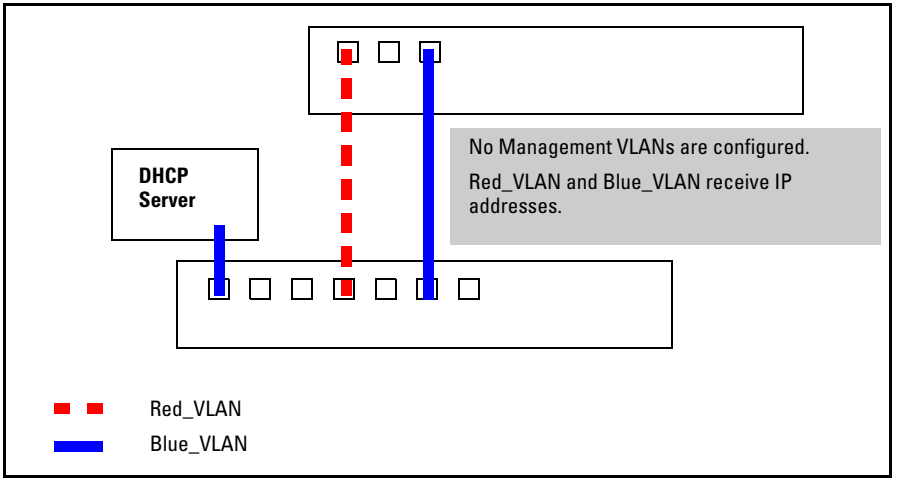


Figure 2-38. Example of no Management VLANs Configured

4. If Red_VLAN is configured as the Management VLAN and the client is on Red_VLAN, but the DHCP server is on Blue_VLAN, the client will not receive an IP address. See figure 2-39.

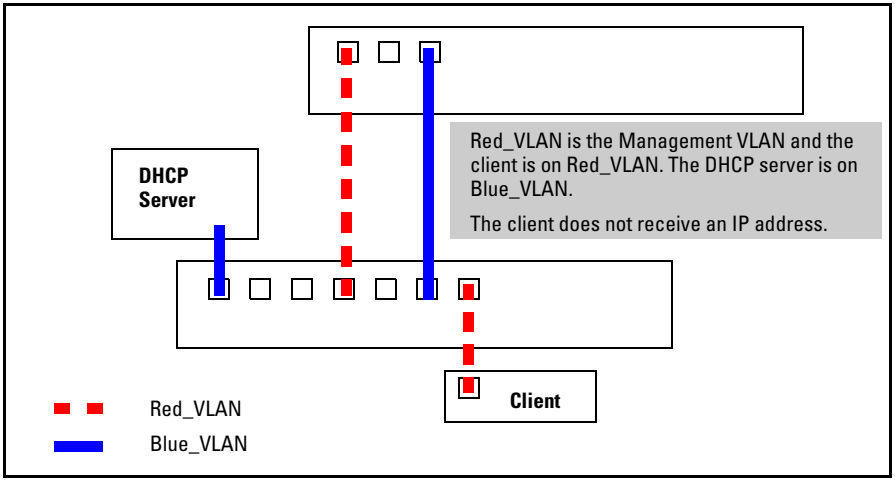

DHCP Server
Red_VLAN is the Management VLAN and the client is on Red_VLAN. The DHCP server is on Blue_VLAN.
The client does not receive an IP address.
Client
Red_VLAN
Blue_VLAN

Figure 2-39. Example of Client on Different Management VLAN from DHCP Server

5. If Blue_VLAN is configured as the Management VLAN, the client is on Blue_VLAN, and the DHCP server is on Blue_VLAN, the client receives an IP address.

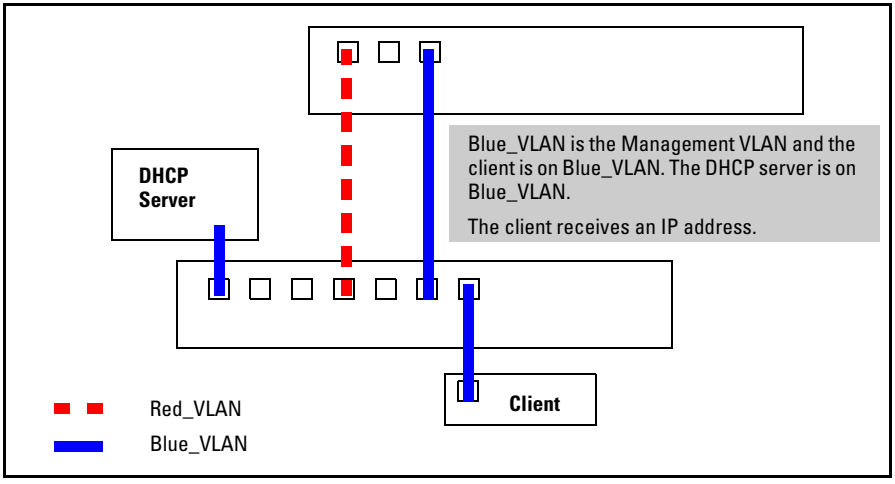

DHCP Server
Blue_VLAN is the Management VLAN and the client is on Blue_VLAN. The DHCP server is on Blue_VLAN.
The client receives an IP address.
Client
Red_VLAN
Blue_VLAN

Figure 2-40. Example of DHCP Server and Client on the Management VLAN

Deleting the Management VLAN

You can disable the Secure Management feature without deleting the VLAN itself. For example, either of the following commands disables the Secure Management feature in the above example:

```
ProCurve (config)# no management-vlan 100
ProCurve (config)# no management-vlan my_vlan
```

Operating Notes for Management VLANs

- Use only a static, port-based VLAN for the Management VLAN.
- The Management VLAN feature applies to both IPv4 and IPv6 traffic.
- The Management VLAN does not support IGMP operation.
- Routing between the Management VLAN and other VLANs is not allowed.
- If there are more than 25 VLANs configured on the switch, reboot the switch after configuring the management VLAN.
- If you implement a Management VLAN in a switch mesh environment, all meshed ports on the switch will be members of the Management VLAN.
- Only one Management-VLAN can be active in the switch. If one Management-VLAN VID is saved in the startup-config file and you configure a different VID in the running-config file, the switch uses the running-config version until you either use the **write-memory** command or reboot the switch.
- During a Telnet session to the switch, if you configure the Management-VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you terminate the session by logging out or rebooting the switch.
- During a web browser session to the switch, if you configure the Management-VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you close the browser session or rebooting the switch.

Note

The Management-VLAN feature does not control management access through a direct connection to the switch's serial port.

- Enabling Spanning Tree where there are multiple links using separate VLANs, including the Management VLAN, between a pair of switches, Spanning Tree will force the blocking of one or more links. This may include the link carrying the Management VLAN, which will cause loss of management access to some devices. This can also occur where meshing is configured and the Management VLAN is configured on a separate link.

- Monitoring Shared Resources: The Management VLAN feature shares internal switch resources with several other features. The switch provides ample resources for all features. However, if the internal resources become fully subscribed, the Management VLAN feature cannot be configured until the necessary resources are released from other uses. For information on determining the current resource availability and usage, refer to the appendix titled "Monitoring Resources" in the *Management and Configuration Guide* for your switch.

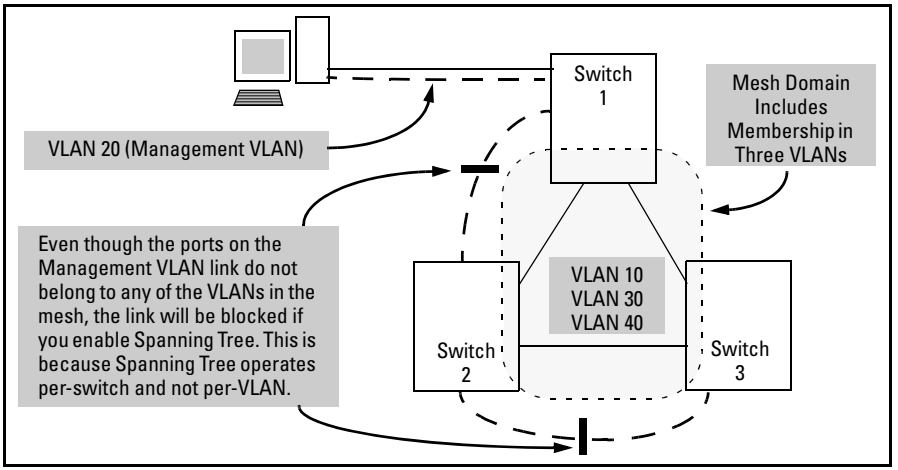

Switch 1
Mesh Domain Includes Membership in Three VLANs
VLAN 20 (Management VLAN)
Even though the ports on the Management VLAN link do not belong to any of the VLANs in the mesh, the link will be blocked if you enable Spanning Tree. This is because Spanning Tree operates per-switch and not per-VLAN.
VLAN 10
VLAN 30
VLAN 40
Switch 2
Switch 3

Figure 2-41. Example of Inadvertently Blocking a Management VLAN Link by Implementing Spanning Tree

Voice VLANs

Configuring voice VLANs separates voice traffic from data traffic and shields your voice traffic from broadcast storms. This section describes how to configure the switch for voice VLAN operation.

Operating Rules for Voice VLANs

- You must statically configure voice VLANs. GVRP and dynamic VLANs do not support voice VLAN operation.
- Configure all ports in a voice VLAN as tagged members of the VLAN. This ensures retention of the QoS (Quality of Service) priority included in voice VLAN traffic moving through your network.
- If a telephone connected to a voice VLAN includes a data port used for connecting other networked devices (such as PCs) to the network, then you must configure the port as a tagged member of the voice VLAN and a tagged or untagged member of the data VLAN you want the other networked device to use.

Components of Voice VLAN Operation

- **Voice VLAN(s):** Configure one or more voice VLANs on the switch. Some reasons for having multiple voice VLANs include:
  - Employing telephones with different VLAN requirements
  - Better control of bandwidth usage
  - Segregating telephone groups used for different, exclusive purposes

  Where multiple voice VLANs exist on the switch, you can use routing to communicate between telephones on different voice VLANs. .
- **Tagged/Untagged VLAN Membership:** If the appliances using a voice VLAN transmit tagged VLAN packets, then configure the member ports as tagged members of the VLAN. Otherwise, configure the ports as untagged members.

Voice VLAN QoS Prioritizing (Optional)

Without configuring the switch to prioritize voice VLAN traffic, one of the following conditions applies:

- If the ports in a voice VLAN are not tagged members, then the switch forwards all traffic on that VLAN at “normal” priority.
- If the ports in a voice VLAN are tagged members, then the switch forwards all traffic on that VLAN at whatever priority the traffic has when received inbound on the switch.

Using the switch’s QoS VLAN-ID (VID) Priority option, you can change the priority of voice VLAN traffic moving through the switch. If all port memberships on the voice VLAN are tagged, the priority level you set for voice VLAN traffic is carried to the next device. With all ports on the voice VLAN configured as tagged members, you can enforce a QoS priority policy moving through the switch and through your network. To set a priority on a voice VLAN, use the following command:

Syntax: vlan < *vid* > qos priority < 0 - 7 >

The qos priority default setting is 0 (normal), with 1 as the lowest priority and 7 as the highest priority.

For example, if you configured a voice VLAN with a VID of 10, and wanted the highest priority for all traffic on this VLAN, you would execute the following command:

```
ProCurve(config) # vlan 10 qos priority 7
ProCurve (config) # write memory
```

Note that you also have the option of resetting the DSCP (DiffServe Code-point) on tagged voice VLAN traffic moving through the switch. For more on this and other QoS topics, refer to the chapter titled "Quality of Service (QoS): Managing Bandwidth More Effectively" in this guide.

Voice VLAN Access Security

You can use port security configured on an individual port or group of ports in a voice VLAN. That is, you can allow or deny access to a phone having a particular MAC address. Refer to chapter titled "Configuring and Monitoring Port Security" in the *Access Security Guide* for your switch.

Note MAC authentication is not recommended in voice VLAN applications.

Effect of VLANs on Other Switch Features

Spanning Tree Operation with VLANs

Depending on the spanning-tree option configured on the switch, the spanning-tree feature may operate as a single instance across all ports on the switch (regardless of VLAN assignments) or multiple instance on a per-VLAN basis. For single-instance operation, this means that if redundant physical links exist between the switch and another 802.1Q device, all but one link will be blocked, regardless of whether the redundant links are in separate VLANs. In this case you can use port trunking to prevent Spanning Tree from unnecessarily blocking ports (and to improve overall network performance). For multiple-instance operation, physically redundant links belonging to different VLANs can remain open. Refer to chapter 4, "Multiple Instance Spanning-Tree Operation" .

Note that Spanning Tree operates differently in different devices. For example, in the (obsolete, non-802.1Q) ProCurve Switch 2000 and the ProCurve Switch 800T, Spanning Tree operates on a per-VLAN basis, allowing redundant physical links as long as they are in separate VLANs.

IP Interfaces

There is a one-to-one relationship between a VLAN and an IP network interface. Since the VLAN is defined by a group of ports, the state (up/down) of those ports determines the state of the IP network interface associated with that VLAN. When a port-based VLAN or an IPv4 or IPv6 protocol-based VLAN comes up because one or more of its ports is up, the IP interface for that VLAN is also activated. Likewise, when a VLAN is deactivated because all of its ports are down, the corresponding IP interface is also deactivated.

VLAN MAC Address

The switches covered by this guide have one unique MAC address for all of their VLAN interfaces. You can send an 802.2 test packet to this MAC address to verify connectivity to the switch. Likewise, you can assign an IP address to the VLAN interface, and when you Ping that address, ARP will resolve the IP address to this single MAC address. In a topology where a switch has multiple VLANs and must be connected to a device having a single forwarding database, such as the Switch 4000M, some cabling restrictions apply. For more on this topic, refer to “Multiple VLAN Considerations” on page 2-20.

Port Trunks

When assigning a port trunk to a VLAN, all ports in the trunk are automatically assigned to the same VLAN. You cannot split trunk members across multiple VLANs. Also, a port trunk is tagged, untagged, or excluded from a VLAN in the same way as for individual, untrunked ports.

Port Monitoring

If you designate a port on the switch for network monitoring, this port will appear in the Port VLAN Assignment screen and can be configured as a member of any VLAN. For information on how broadcast, multicast, and unicast packets are tagged inside and outside of the VLAN to which the monitor port is assigned, refer to the section titled “VLAN-Related Problems” in the “Troubleshooting” appendix of the *Management and Configuration Guide* for your switch.

Jumbo Packet Support

Jumbo packet support is enabled per-VLAN and applies to all ports belonging to the VLAN. For more information, refer to the chapter titled “Port Traffic Controls” in the *Management and Configuration Guide* for your switch.

VLAN Restrictions

- A port must be a member of at least one VLAN. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN; VID = 1).
- A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged. (The "Untagged" designation enables VLAN operation with non 802.1Q-compliant devices.)
- A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing the same type, the port can be an untagged member of only one such VLAN.
- With routing enabled on the switch, the switch can route traffic between:
  - Multiple, port-based VLANs
  - A port-based VLAN and an IPv4 protocol-based VLAN
  - A port-based VLAN and an IPv6 protocol-based VLAN
  - An IPv4 protocol-based VLAN and an IPv6 protocol VLAN.

  Other, routable, protocol-based VLANs must use an external router to move traffic between VLANs. With routing disabled, all routing between VLANs must be through an external router.
- Prior to deleting a static VLAN, you must first re-assign all ports in the VLAN to another VLAN. You can use the **no vlan < *vid* >** command to delete a static VLAN. For more information, refer to "Creating a New Static VLAN (Port-Based or Protocol-Based) Changing the VLAN Context Level" on page 2-41.

Migrating Layer 3 VLANs Using VLAN MAC Configuration

ProCurve routing switches provide an easy way to maintain Layer 3 VLAN configurations when you migrate distribution routers in a network configuration that is not centrally managed. By following the procedure described in this section, you can upgrade to ProCurve routing switches without stopping the operation of attached hosts that use existing routers as their default gateway to route traffic between VLANs. You can achieve seamless VLAN migration by configuring the MAC address of the previously installed router on the VLAN interfaces of a ProCurve routing switch.

VLAN MAC Address Reconfiguration

The ProCurve switches covered by this guide use one unique MAC address for all VLAN interfaces. If you assign an IP address to a VLAN interface, ARP resolves the IP address to the MAC address of the routing switch for all incoming packets.

The Layer 3 VLAN MAC Configuration feature allows you to reconfigure the MAC address used for VLAN interfaces using the CLI. Packets addressed to the reconfigured Layer 3 MAC address, such as ARP and IP data packets, are received and processed by the ProCurve routing switch.

Packets transmitted from the routing switch (packets originating from the router and forwarded packets) use the original ProCurve MAC address as the source MAC address in Ethernet headers.

ARP reply packets use the reconfigured MAC address in both the:

- ARP Sender MAC address field.
- Source MAC address field in the Ethernet frame header

When you reconfigure the MAC address on a VLAN interface, you may also specify a keepalive timeout to transmit heartbeat packets that advertise the new MAC address.

By configuring the MAC address of the previously installed router as the MAC address of each VLAN interface on a ProCurve switch, you can swap the physical port of a router to the ProCurve switch after the switch has been properly configured in the network.

Handling Incoming and Outgoing VLAN Traffic

Incoming VLAN data packets and ARP requests are received and processed on the routing switch according to the MAC address of the previously installed router that is configured for each VLAN interface.

Outgoing VLAN traffic uses the MAC address of the ProCurve switch as the source MAC address in packet headers. The MAC address configured on VLAN interfaces is not used on outbound VLAN traffic.

When the routing switch receives an ARP request for the IP address configured on a VLAN interface, the ARP reply uses the reconfigured MAC address in both the:

- ARP Sender MAC address field
- Source MAC address field in the Ethernet frame header.

When proxy ARP is enabled on a VLAN interface, the "gracious" ARP reply sent for an ARP request received from VLAN devices located outside the directly connected IP subnets also contains the reconfigured MAC address in the:

- ARP Sender MAC address field
- Source MAC address field in the Ethernet frame header.

Note

The Virtual Router Redundancy Protocol (VRRP) is not supported on VLAN interfaces on which the MAC address for incoming traffic has been reconfigured

To hosts in the network, VLAN traffic continues to be routed (using the reconfigured MAC address as destination address), but outbound VLAN traffic appears to be sent from another router (using the ProCurve MAC address as source address) attached to the same subnet. Although it appears as an asymmetric path to network hosts, the MAC address configuration feature enables Layer 3 VLAN migration. (A successful VLAN migration is achieved because the hosts do not verify that the source MAC address and the destination MAC address are the same when communicating with the routing switch.)

Sending Heartbeat Packets with a Configured MAC Address

On the VLAN interfaces of a routing switch, the user-defined MAC address only applies to inbound traffic. As a result, any connected switches need to learn the new address that is included in the Ethernet frames of outbound VLAN traffic transmitted from the routing switch.

If a connected switch does not have the newly configured MAC address of the routing switch as a destination in its MAC address table, it floods packets to all of its ports until a return stream allows the switch to learn the correct destination address. As a result, the performance of the switch is degraded as it tries to send Ethernet packets to an unknown destination address.

To allow connected switches to learn the user-configured MAC address of a VLAN interface, the ProCurve routing switch can send periodic heartbeat-like Ethernet packets. The Ethernet packets contain the configured MAC address as the source address in the packet header. IP multicast packets or Ethernet service frames are preferred because they do not interrupt the normal operation of client devices connected on the segment.

Because the aging time of destination addresses in MAC address tables varies on network devices, you must also configure a time interval to use for sending heartbeat packets.

Heartbeat packets are sent at periodic intervals with a specific ProCurve unicast MAC address in destination field. This MAC address is assigned to ProCurve and is not used by other non-ProCurve routers. Because the heartbeat packet contains a unicast MAC address, it does not interrupt host operation. Even if you have multiple ProCurve switches connected to the network, there is no impact on network performance because each switch sends heartbeat packets with its configured MAC address as the destination address.

The format of a heartbeat packet is an extended Ethernet OUI frame with an extended OUI Ethertype (88B7) and a new protocol identifier in the 5-octet protocol identifier field.

Configuring a VLAN MAC Address with Heartbeat Interval

When installing ProCurve routing switches in the place of existing routers in a network configuration, you can achieve Layer 3 VLAN migration by using the **ip-recv-mac-address** command at the VLAN configuration level to:

- Configure the MAC address of the previously installed router on each VLAN interface of a ProCurve routing switch.
- Optionally configure the time interval to use for sending heartbeat packets with the configured MAC address.

Syntax: [no] ip-recv-mac-address <*mac-address* > [interval <*seconds*>]

ip-recv-mac-address <*mac-address*>

Configures a VLAN interface with the specified MAC address. Enter the **no** *version of the command to remove the configured MAC address and return to the original MAC address of the ProCurve switch.*

interval <*seconds*>

(Optional) Configures the time interval (in seconds) used between transmissions of heartbeat packets to all network devices configured on the VLAN. Valid values are from one to 255 seconds. The default is 60 seconds.

Operating Notes

- The **ip-recv-mac-address** command allows you to configure only one MAC address for a specified VLAN. If you re-enter the command to configure another MAC address, the previously configured MAC address is overwritten.
- Enter the **no** form of the command to remove a configured MAC address and restore the default MAC address of the ProCurve switch.
- When you configure a VLAN MAC address, you may also specify a heartbeat interval. The **interval** <*seconds*> parameter is optional.
- After you configure a VLAN MAC address:
  - IP router and MAC ARP replies to other VLAN devices contain the user-defined MAC address as the Ethernet sender hardware address.
  - Outbound VLAN traffic contains the ProCurve MAC address, not the configured MAC address, as the source MAC address in packet headers.

- Immediately after you configure a VLAN MAC address or remove a configured MAC address, a gratuitous ARP message is broadcast on the connected segment to announce the change of the IP-to-MAC address binding to all connected IP-based equipment.
- A configured VLAN MAC address supports proxy ARP and gracious ARP.
- A new MIB variable, **ifRcvAddressTable**, is introduced to support VLAN MAC configuration.
- You cannot configure a VLAN MAC address using the web browser or menu interface. You must use the CLI.
- VRRP is not supported on a VLAN interface with a user-configured MAC address.

Example

The following example shows how to configure a MAC address on VLAN 101.

```
ProCurve# configure terminal
ProCurve(config)# vlan 101
ProCurve(vlan-101)# ip-recv-mac-address 0060b0-e9a200
interval 100
```

Verifying a VLAN MAC Address Configuration

To verify the configuration of Layer 3 MAC addresses on the VLAN interfaces of a switch, enter the **show ip-recv-mac-address** command.

```
ProCurve# show ip-recv-mac-address

 VLAN L3-Mac-Address Table

 VLAN            L3-Mac-Address  Timeout
 ------------    ----------------------   ----------
 DEFAULT_VLAN  001635-024467    60
 VLAN2         001635-437529    100
```

Figure 2-42. Example of Displaying a VLAN MAC Address

GVRP

Contents

Overview 3-2

Introduction 3-3

General Operation 3-4

Per-Port Options for Handling GVRP "Unknown VLANs" 3-7

Per-Port Options for Dynamic VLAN Advertising and Joining 3-9

GVRP and VLAN Access Control 3-11

Advertisements and Dynamic Joins 3-11

Port-Leave From a Dynamic VLAN 3-11

Planning for GVRP Operation 3-12

Configuring GVRP On a Switch 3-13

Menu: Viewing and Configuring GVRP 3-13

CLI: Viewing and Configuring GVRP 3-14

Web: Viewing and Configuring GVRP 3-18

GVRP Operating Notes 3-18

Overview

This chapter describes GVRP and how to configure it with the switch's built-in interfaces, and assumes an understanding of VLANs, which are described in chapter 2, "Static Virtual LANs (VLANs)" .

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

- Chapter 3, "Using the Menu Interface"
- Chapter 4, "Using the Command Line Interface (CLI)"
- Chapter 5, "Using the Web Browser Interface
- Chapter 6, "Switch Memory and Configuration"

Introduction

Feature	Default	Menu	CLI	Web
view GVRP configuration	n/a	page 3-13	page 3-14	page 3-18
list static and dynamic VLANs on a GVRP-enabled switch	n/a	—	page 3-16	page 3-18
enable or disable GVRP	disabled	page 3-13	page 3-15	page 3-18
enable or disable GVRP on individual ports	enabled	page 3-13	page 3-15	—
control how individual ports handle advertisements for new VLANs	Learn	page 3-13	page 3-15	page 3-18
convert a dynamic VLAN to a static VLAN	n/a	—	page 3-17	—
configure static VLANs	DEFAULT_VLAN (VID = 1)	page 2-24	page 2-30	page 2-45

GVRP—GARP VLAN Registration Protocol—is an application of the Generic Attribute Registration Protocol—GARP. GVRP is defined in the IEEE 802.1Q standard, and GARP is defined in the IEEE 802.1D-1998 standard.

Note

To understand and use GVRP you must have a working knowledge of 802.1Q VLAN tagging. (Refer to chapter 2, “Static Virtual LANs (VLANs)” .)

GVRP uses “GVRP Bridge Protocol Data Units” (“GVRP BPDUs”) to “advertise” static VLANs. In this manual, a GVRP BPDU is termed an *advertisement*. Advertisements are sent outbound from ports on a switch to the devices directly connected to those ports.

While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.

GVRP enables the switch to dynamically create 802.1Q-compliant VLANs on links with other devices running GVRP. This enables the switch to automatically create VLAN links between GVRP-aware devices. (A GVRP link can include intermediate devices that are not GVRP-aware.) This operation reduces the chances for errors in VLAN configuration by automatically providing VLAN ID (VID) consistency across the network. That is, you can use GVRP to propagate VLANs to other GVRP-aware devices instead of manually

having to set up VLANs across your network. After the switch creates a dynamic VLAN, you can optionally use the CLI **static <*vlan-id*>** command to convert it to a static VLAN or allow it to continue as a dynamic VLAN for as long as needed. You can also use GVRP to dynamically enable port membership in static VLANs configured on a switch.

Note: On the switches covered in this guide, GVRP can be enabled only if **max vlans** is set to no more than 256 VLANs.

General Operation

When GVRP is enabled on a switch, the VID for any static VLANs configured on the switch is *advertised* (using BPDUs—Bridge Protocol Data Units) out all ports, regardless of whether a port is up or assigned to any particular VLAN. A GVRP-aware port on another device that receives the advertisements over a link can dynamically join the advertised VLAN.

A dynamic VLAN (that is, a VLAN learned through GVRP) is tagged on the port on which it was learned. Also, a GVRP-enabled port can forward an advertisement for a VLAN it learned about from other ports on the same switch (internal source), but the forwarding port will not itself join that VLAN until an advertisement for that VLAN is received through a link from another device (external source) on that specific port

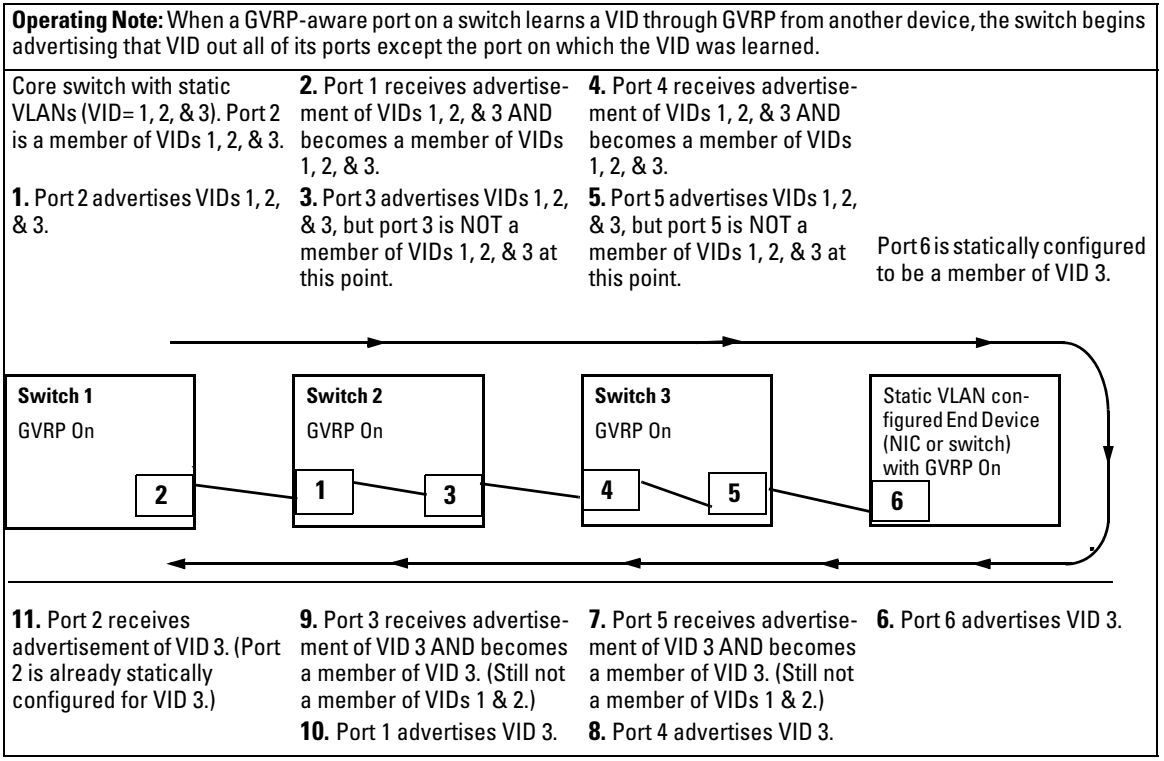

Operating Note: When a GVRP-aware port on a switch learns a VID through GVRP from another device, the switch begins advertising that VID out all of its ports except the port on which the VID was learned.
Core switch with static VLANs (VID= 1, 2, & 3). Port 2 is a member of VIDs 1, 2, & 3.
1. Port 2 advertises VIDs 1, 2, & 3.
2. Port 1 receives advertisement of VIDs 1, 2, & 3 AND becomes a member of VIDs 1, 2, & 3.
3. Port 3 advertises VIDs 1, 2, & 3, but port 3 is NOT a member of VIDs 1, 2, & 3 at this point.
4. Port 4 receives advertisement of VIDs 1, 2, & 3 AND becomes a member of VIDs 1, 2, & 3.
5. Port 5 advertises VIDs 1, 2, & 3, but port 5 is NOT a member of VIDs 1, 2, & 3 at this point.
Port 6 is statically configured to be a member of VID 3.
Switch 1
GVRP On
Switch 2
GVRP On
Switch 3
GVRP On
Static VLAN configured End Device (NIC or switch) with GVRP On
2
1
3
4
5
6
11. Port 2 receives advertisement of VID 3. (Port 2 is already statically configured for VID 3.)
9. Port 3 receives advertisement of VID 3 AND becomes a member of VID 3. (Still not a member of VIDs 1 & 2.)
10. Port 1 advertises VID 3.
7. Port 5 receives advertisement of VID 3 AND becomes a member of VID 3. (Still not a member of VIDs 1 & 2.)
8. Port 4 advertises VID 3.
6. Port 6 advertises VID 3.

Figure 3-1. .Example of Forwarding Advertisements and Dynamic Joining

Note that if a static VLAN is configured on at least one port of a switch, and that port has established a link with another device, then all other ports of that switch will send advertisements for that VLAN.

For example, in the following figure, Tagged VLAN ports on switch “A” and switch “C” advertise VLANs 22 and 33 to ports on other GVRP-enabled switches that can dynamically join the VLANs.

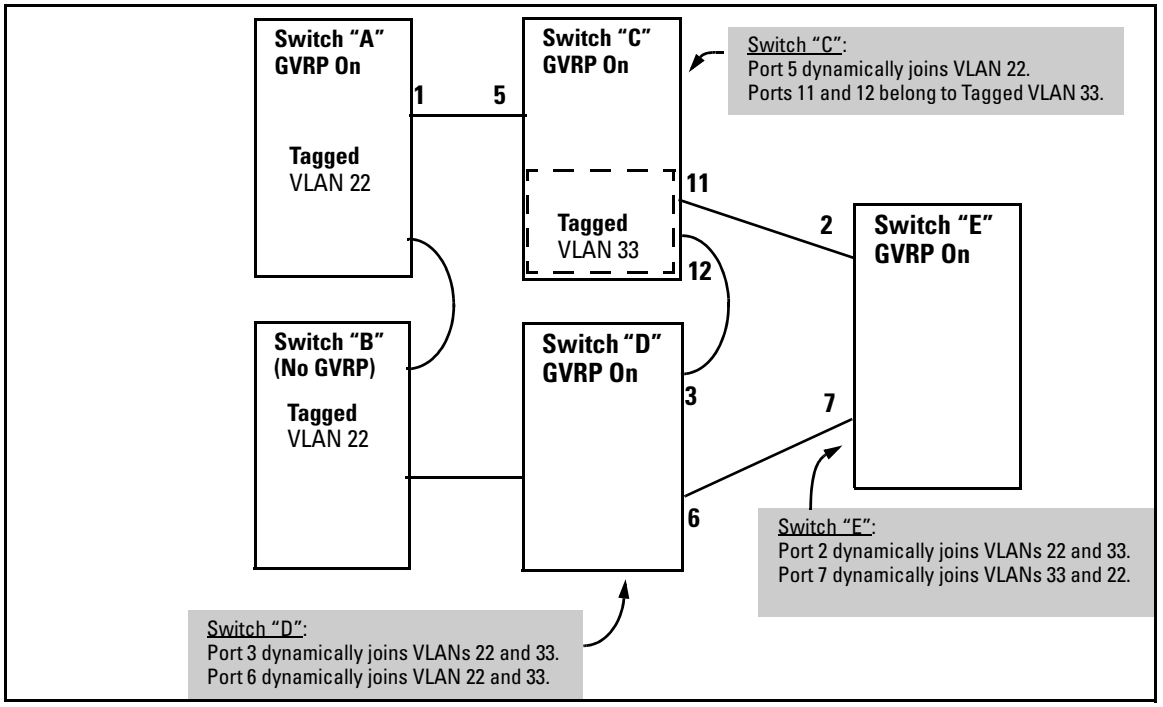


Figure 3-2. Example of GVRP Operation

Note

A port can learn of a dynamic VLAN through devices that are not aware of GVRP (Switch "B", above). VLANs must be disabled in GVRP-unaware devices to allow tagged packets to pass through.

A GVRP-aware port receiving advertisements has these options:

- If there is not already a static VLAN with the advertised VID on the receiving port, then dynamically create the VLAN and become a member.
- If the switch already has a static VLAN assignment with the same VID as in the advertisement, and the port is configured to **Auto** for that VLAN, then the port will dynamically join the VLAN and begin moving that VLAN's traffic. (For more detail on **Auto**, see "Per-Port Options for Dynamic VLAN Advertising and Joining" on page 3-9.)
- Ignore the advertisement for that VID.
- Don't participate in that VLAN.

Note also that a port belonging to a Tagged or Untagged static VLAN has these configurable options:

- Send VLAN advertisements, and also receive advertisements for VLANs on other ports and dynamically join those VLANs.
- Send VLAN advertisements, but ignore advertisements received from other ports.
- Avoid GVRP participation by not sending advertisements and dropping any advertisements received from other devices.

IP Addressing. A dynamic VLAN does not have an IP address, and moves traffic on the basis of port membership in VLANs. However, after GVRP creates a dynamic VLAN, you can convert it to a static VLAN. Note that it is then necessary to assign ports to the VLAN in the same way that you would for a static VLAN that you created manually. In the static state you can configure IP addressing on the VLAN and access it in the same way that you would any other static (manually created) VLAN.

Per-Port Options for Handling GVRP "Unknown VLANs"

An "unknown VLAN" is a VLAN that the switch learns of by receiving an advertisement for that VLAN on a port that is not already a member of that VLAN. If the port is configured to learn unknown VLANs, then the VLAN is dynamically created and the port becomes a tagged member of the VLAN. For example, suppose that in figure 3-2 (page 3-6), port 1 on switch "A" is connected to port 5 on switch "C". Because switch "A" has VLAN 22 statically configured, while switch "C" does not have this VLAN statically configured (and does not "Forbid" VLAN 22 on port 5), VLAN 22 is handled as an "Unknown VLAN" on port 5 in switch "C". Conversely, if VLAN 22 was statically configured on switch C, but port 5 was not a member, port 5 would become a member when advertisements for VLAN 22 were received from switch "A".

When you enable GVRP on a switch, you have the per-port join-request options listed in table 3-1:

Table 3-1. Options for Handling "Unknown VLAN" Advertisements:

Unknown VLAN Mode	Operation
Learn (the Default)	Enables the port to become a member of any unknown VLAN for which it receives an advertisement. Allows the port to advertise other VLANs that have at least one other port on the same switch as a member.
Block	Prevents the port from joining any new dynamic VLANs for which it receives an advertisement. Allows the port to advertise other VLANs that have at least one other port as a member.
Disable	Causes the port to ignore and drop all GVRP advertisements it receives and also prevents the port from sending any GVRP advertisements.

The CLI **show gvrp** command and the menu interface VLAN Support screen show a switch's current GVRP configuration, including the Unknown VLAN settings.

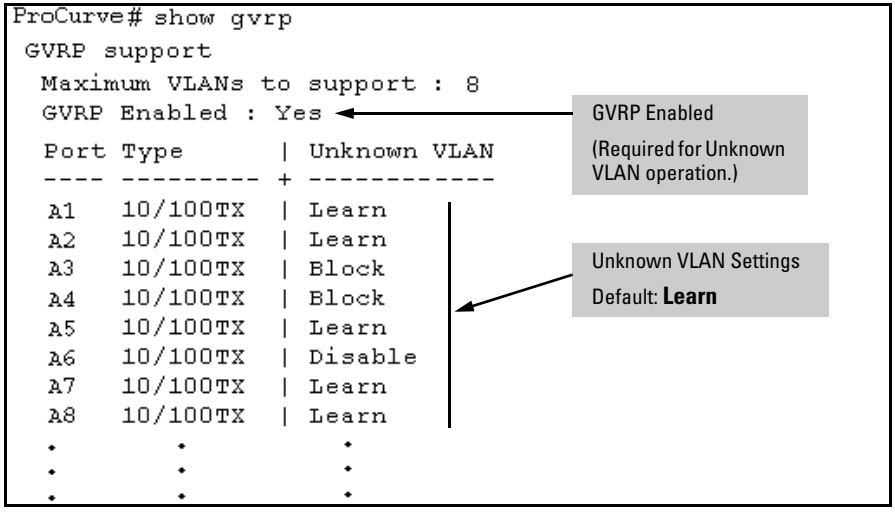


Figure 3-3. Example of GVRP Unknown VLAN Settings

Per-Port Options for Dynamic VLAN Advertising and Joining

Initiating Advertisements. As described in the preceding section, to enable dynamic joins, GVRP must be enabled and a port must be configured to Learn (the default). However, to send advertisements in your network, one or more static (**Tagged**, **Untagged**, or **Auto**) VLANs must be configured on one or more switches (with GVRP enabled), depending on your topology.

Enabling a Port for Dynamic Joins. You can configure a port to dynamically join a static VLAN. The join will then occur if that port subsequently receives an advertisement for the static VLAN. (This is done by using the **Auto** and **Learn** options described in table 3-2, on the next page.

Parameters for Controlling VLAN Propagation Behavior. You can configure an individual port to actively or passively participate in dynamic VLAN propagation or to ignore dynamic VLAN (GVRP) operation. These options are controlled by the GVRP "Unknown VLAN" and the static VLAN configuration parameters, as described in the following table:

Table 3-2. Controlling VLAN Behavior on Ports with Static VLANs

Per-Port "Unknown VLAN" (GVRP) Configuration	Static VLAN Options—Per VLAN Specified on Each Port [1]		
	Port Activity: Tagged or Untagged (Per VLAN)[2]	**Port Activity: Auto[2] (Per VLAN)**	**Port Activity: Forbid (Per VLAN)[2]**
Learn (the Default)	The port: • Belongs to specified VLAN. • Advertises specified VLAN. • Can become a member of dynamic VLANs for which it receives advertisements. • Advertises dynamic VLANs that have at least one other port (on the same switch) as a member.	The port: • Will become a member of specified VLAN if it receives advertisements for specified VLAN from another device. • Will advertise specified VLAN. • Can become a member of other, dynamic VLANs for which it receives advertisements. • Will advertise a dynamic VLAN that has at least one other port (on the same switch) as a member.	The port: 1. Will not become a member of the specified VLAN. 2. Will not advertise specified VLAN. 3. Can become a member of other dynamic VLANs for which it receives advertisements. 4. Will advertise a dynamic VLAN that has at least one other port on the same switch as a member.
Block	The port: • Belongs to the specified VLAN. • Advertises this VLAN. • Will not become a member of new dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port as a member.	The port: • Will become a member of specified VLAN if it receives advertisements for this VLAN. • Will advertise this VLAN. • Will not become a member of new dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port (on the same switch) as a member.	The port: • Will not become a member of the specified VLAN. • Will not advertise this VLAN. • Will not become a member of dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port (on the same switch) as a member.
Disable	The port: • Is a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any advertised VLANs. • Will not advertise VLANs.	The port: • Will not become a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.	The port: • Will not become a member of this VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.

[1] Each port of the switch must be a Tagged or Untagged member of at least one VLAN. Thus, any port configured for GVRP to Learn or Block will generate and forward advertisements for static VLAN(s) configured on the switch and also for dynamic VLANs the switch learns on other ports.

[2] To configure tagging, **Auto**, or **Forbid**, see "Configuring Static VLAN Per-Port Settings" on page 2-43 (for the CLI) or "Adding or Changing a VLAN Port Assignment" on page 2-28 (for the menu).

As the preceding table indicates, when you enable GVRP, a port that has a Tagged or Untagged static VLAN has the option for both generating advertisements and dynamically joining other VLANs.

Note

In table 3-2, above, the Unknown VLAN parameters are configured on a per-port basis using the CLI. The Tagged, Untagged, Auto, and Forbid options are configured per static VLAN on every port, using either the menu interface or the CLI.

Because dynamic VLANs operate as Tagged VLANs, and because a tagged port on one device cannot communicate with an untagged port on another device, ProCurve recommends that you use Tagged VLANs for the static VLANs you will use to generate advertisements.

GVRP and VLAN Access Control

Advertisements and Dynamic Joins

When you enable GVRP on a switch, the default GVRP parameter settings allow all of the switch's ports to transmit and receive dynamic VLAN advertisements (GVRP advertisements) and to dynamically join VLANs. The two preceding sections describe the per-port features you can use to control and limit VLAN propagation. To summarize, you can:

- Allow a port to advertise and/or join dynamic VLANs (Learn mode—the default).
- Allow a port to send VLAN advertisements, but not receive them from other devices; that is, the port cannot dynamically join a VLAN but other devices can dynamically join the VLANs it advertises (Block mode).
- Prevent a port from participating in GVRP operation (Disable mode).

Port-Leave From a Dynamic VLAN

A dynamic VLAN continues to exist on a port for as long as the port continues to receive advertisements of that VLAN from another device connected to that port or until you:

- Convert the VLAN to a static VLAN (See "Converting a Dynamic VLAN to a Static VLAN" on page 3-17.)
- Reconfigure the port to **Block** or **Disable**

- Disable GVRP
- Reboot the switch

The time-to-live for dynamic VLANs is 10 seconds. That is, if a port has not received an advertisement for an existing dynamic VLAN during the last 10 seconds, the port removes itself from that dynamic VLAN.

Planning for GVRP Operation

These steps outline the procedure for setting up dynamic VLANs for a segment.

1. Determine the VLAN topology you want for each segment (broadcast domain) on your network.
2. Determine the VLANs that must be static and the VLANs that can be dynamically propagated.
3. Determine the device or devices on which you must manually create static VLANs in order to propagate VLANs throughout the segment.
4. Determine security boundaries and how the individual ports in the segment will handle dynamic VLAN advertisements. (See table 3-1 on page 3-8 and table 3-2 on page 3-10.)
5. Enable GVRP on all devices you want to use with dynamic VLANs and configure the appropriate "Unknown VLAN" parameter (**Learn**, **Block**, or **Disable**) for each port.
6. Configure the static VLANs on the switch(es) where they are needed, along with the per-VLAN parameters (**Tagged**, **Untagged**, **Auto**, and **Forbid**—see table 3-2 on page 3-10) on each port.
7. Dynamic VLANs will then appear automatically, according to the configuration options you have chosen.
8. Convert dynamic VLANs to static VLANs where you want dynamic VLANs to become permanent.

Configuring GVRP On a Switch

The procedures in this section describe how to:

- View the GVRP configuration on a switch
- Enable and disable GVRP on a switch
- Specify how individual ports will handle advertisements

To view or configure static VLANs for GVRP operation, refer to "Per-Port Static VLAN Configuration Options" on page 2-14.

Menu: Viewing and Configuring GVRP

1. From the Main Menu, select:

   2. Switch Configuration ...
   8. VLAN Menu ...
   1. VLAN Support

```
==========================- CONSOLE - MANAGER MODE -============================
                    Switch Configuration - VLAN - VLAN Support

  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : No


 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 3-4. The VLAN Support Screen (Default Configuration)

2. Do the following to enable GVRP and display the Unknown VLAN fields:
   a. Press **[E]** (for **Edit**).
   b. Use ↓ to move the cursor to the **GVRP Enabled** field.
   c. Press the Space bar to select **Yes**.
   d. Press ↓ again to display the **Unknown VLAN** fields.

The Unknown VLAN fields enable you to configure each port to:
- Learn - Dynamically join any advertised VLAN and advertise all VLANs learned through other ports.
- Block - Do not dynamically join any VLAN, but still advertise all VLANs learned through other ports.
- Disable - Ignore and drop all incoming advertisements and do not transmit any advertisements.

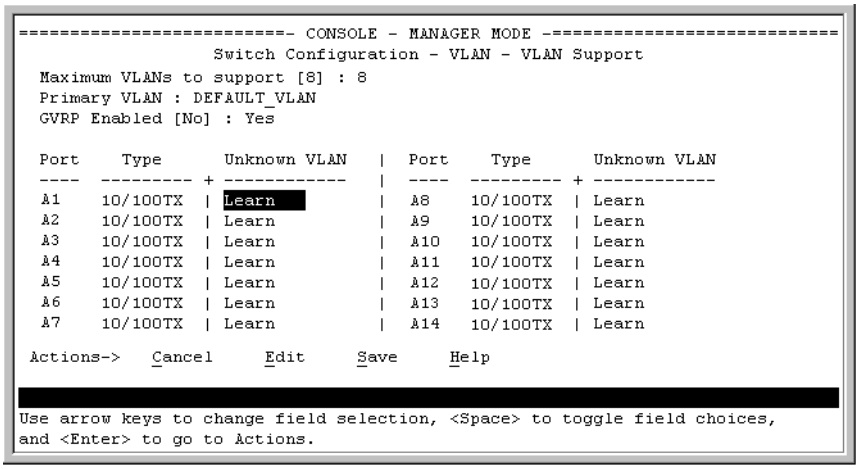

========== CONSOLE - MANAGER MODE ==========
Switch Configuration - VLAN - VLAN Support
Maximum VLANs to support [8] : 8
Primary VLAN : DEFAULT_VLAN
GVRP Enabled [No] : Yes
Port Type Unknown VLAN | Port Type Unknown VLAN
A1 10/100TX Learn | A8 10/100TX Learn
A2 10/100TX Learn | A9 10/100TX Learn
A3 10/100TX Learn | A10 10/100TX Learn
A4 10/100TX Learn | A11 10/100TX Learn
A5 10/100TX Learn | A12 10/100TX Learn
A6 10/100TX Learn | A13 10/100TX Learn
A7 10/100TX Learn | A14 10/100TX Learn
Actions-> Cancel Edit Save Help
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.

Figure 3-5. Example Showing Default Settings for Handling Advertisements

3. Use the arrow keys to select the port you want, and the Space bar to select Unknown VLAN option for any ports you want to change.
4. When you finish making configuration changes, press **[Enter]**, then **[S]** (for **Save**) to save your changes to the Startup-Config file.

CLI: Viewing and Configuring GVRP

GVRP Commands Used in This Section

show gvrp	below
gvrp	page 3-15
unknown-vlans	page 3-15

Displaying the Switch's Current GVRP Configuration. This command shows whether GVRP is disabled, along with the current settings for the maximum number of VLANs and the current Primary VLAN. (For more on the last two parameters, see chapter 2, "Static Virtual LANs (VLANs)" .)

Syntax: show gvrp *Shows the current settings.*

```
ProCurve> show gvrp
 GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : No
```

Figure 3-6. Example of "Show GVRP" Listing with GVRP Disabled

```
ProCurve> show gvrp
 GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : Yes

  Port Type      | Unknown VLAN
  ---- --------- + ------------
 A1    10/100TX  | Learn
 A2    10/100TX  | Learn
 A3    10/100TX  | Block
 A4    10/100TX  | Disable
 A5    10/100TX  | Disable
 A6    10/100TX  | Learn
 A7    10/100TX  | Learn
  .       .      |   .
  .       .      |   .
  .       .      |   .
```

This example includes non-default settings for the Unknown VLAN field for some ports.

Figure 3-7. Example of Show GVRP Listing with GVRP Enabled

Enabling and Disabling GVRP on the Switch. This command enables GVRP on the switch.

Syntax: gvrp

This example enables GVRP:

```
ProCurve(config)# gvrp
```

This example disables GVRP operation on the switch:

```
ProCurve(config)# no gvrp
```

Enabling and Disabling GVRP On Individual Ports. When GVRP is enabled on the switch, use the **unknown-vlans** command to change the Unknown VLAN field for one or more ports. You can use this command at either the Manager level or the interface context level for the desired port(s).

Syntax: interface < *port-list* > unknown-vlans < learn | block | disable >

Changes the Unknown VLAN field setting for the specified port(s).

For example, to change and view the configuration for ports A1-A2 to **Block**:

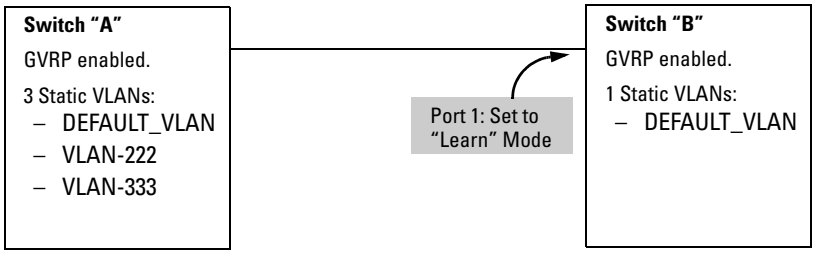
```
ProCurve(config)interface a1-a2 unknown-vlans block

HP4108(config)show gvrp
GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : Yes

  Port Type       | Unknown VLAN
  ---- ---------- + ------------
  1    10/100TX   | Block
  2    10/100TX   | Block
  3    10/100TX   | Learn
  4    10/100TX   | Learn
  .       .           .
  .       .           .
  .       .           .
```

Figure 3-8. Displaying the Static and Dynamic VLANs Active on the Switch

Syntax: show vlans

The **show vlans** *command lists all VLANs present in the switch.*

For example, in the following illustration, switch "B" has one static VLAN (the default VLAN), with GVRP enabled and port 1 configured to **Learn** for Unknown VLANs. Switch "A" has GVRP enabled and has three static VLANs: the default VLAN, VLAN-222, and VLAN-333. In this scenario, switch B will dynamically join VLAN-222 and VLAN-333:



The **show vlans** command lists the dynamic (and static) VLANs in switch "B" after it has learned and joined VLAN-222 and VLAN-333.

```
Switch-B> show vlans

   Status and Counters - VLAN Information

    VLAN support : Yes
    Maximum VLANs to support : 8
    Primary VLAN : DEFAULT_VLAN

    802.1Q VLAN ID Name           Status
    -------------- -------------- ----------
    1              DEFAULT_VLAN   Static
    222            GVRP_222       Dynamic
    333            GVRP_333       Dynamic
```

Dynamic VLANs Learned from Switch "A" through Port 1

Figure 3-9. Example of Listing Showing Dynamic VLANs

Converting a Dynamic VLAN to a Static VLAN. If a port on the switch has joined a dynamic VLAN, you can use the following command to convert that dynamic VLAN to a static VLAN:

Syntax: static < *dynamic-vlan-id* >

Converts the a dynamic VLAN to a static VLAN.

For example, to convert dynamic VLAN 333 (from the previous example) to a static VLAN:

```
ProCurve(config)# static 333
```

When you convert a dynamic VLAN to a static VLAN, all ports on the switch are assigned to the VLAN in Auto mode.

Web: Viewing and Configuring GVRP

To view, enable, disable, or reconfigure GVRP:

1. Click on the **Configuration** tab.
2. Click on **[VLAN Configuration]** and do the following:
   - To enable or disable GVRP, click on **GVRP Enabled**.
   - To change the Unknown VLAN field for any port:
     i. Click on **[GVRP Security]** and make the desired changes.
     ii. Click on **[Apply]** to save and implement your changes to the Unknown VLAN fields.

For web-based Help on how to use the web browser interface screen, click on the **[?]** button provided on the web browser screen.

GVRP Operating Notes

- A dynamic VLAN must be converted to a static VLAN before it can have an IP address.
- On the switches covered in this guide, GVRP can be enabled only if **max vlans** is set to no more than 256 VLANs.
- The total number of VLANs on the switch (static and dynamic combined) cannot exceed the current Maximum VLANs setting. For example, in the factory default state, the switch supports up to 256 VLANs. Any additional VLANs advertised to the switch will not be added unless you first increase the Maximum VLANs setting. In the Menu interface, click on **2. Switch Configuration ... | 8. VLAN Menu | 1. VLAN Support**. In the global config level of the CLI, use **max-vlans**.
- Converting a dynamic VLAN to a static VLAN and then executing the **write memory** command saves the VLAN in the startup-config file and makes it a permanent part of the switch's VLAN configuration.
- Within the same broadcast domain, a dynamic VLAN can pass through a device that is not GVRP-aware. This is because a hub or a switch that is not GVRP-ware will flood the GVRP (multicast) advertisement packets out all ports.
- GVRP assigns dynamic VLANs as Tagged VLANs. To configure the VLAN as Untagged, you must first convert it to a static VLAN.

- Rebooting a switch on which a dynamic VLAN exists deletes that VLAN. However, the dynamic VLAN re-appears after the reboot if GVRP is enabled and the switch again receives advertisements for that VLAN through a port configured to add dynamic VLANs.
- By receiving advertisements from other devices running GVRP, the switch learns of static VLANs on those other devices and dynamically (automatically) creates tagged VLANs on the links to the advertising devices. Similarly, the switch advertises its static VLANs to other GVRP-aware devices, as well as the dynamic VLANs the switch has learned.
- A GVRP-enabled switch does not advertise any GVRP-learned VLANs out of the port(s) on which it originally learned of those VLANs.
- While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.
- A VLAN enabled for jumbo traffic cannot be used to create a dynamic VLAN. A port belonging to a statically configured, jumbo-enabled VLAN cannot join a dynamic VLAN.

4

Multiple Instance Spanning-Tree Operation

Contents

Overview 4-3

802.1s Multiple Spanning Tree Protocol (MSTP) 4-6

MSTP Structure 4-7

How MSTP Operates 4-9

MST Regions 4-9

Regions, Legacy STP and RSTP Switches, and the Common Spanning Tree (CST) 4-11

MSTP Operation with 802.1Q VLANs 4-12

Terminology 4-13

Operating Rules 4-15

MSTP Compatibility with RSTP or STP 4-16

Configuring MSTP 4-17

Planning an MSTP Application 4-17

MSTP Configuration Overview 4-19

Configuring MSTP Operation Mode and Global Settings 4-21

Configuring MSTP Per-Port Parameters 4-26

Configuring Per Port Parameters 4-27

Configuring BPDU Filtering 4-30

Configuring BPDU Protection 4-31

PVST Protection and Filtering 4-34

Configuring MST Instance Parameters 4-39

Configuring MST Instance Per-Port Parameters 4-41

Enabling or Disabling Spanning Tree Operation 4-44

Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another 4-44

MSTP VLAN Configuration Enhancement 4-46

PreConfiguring VLANs in an MST Instance 4-47

Configuring MSTP Instances with the VLAN Range Option 4-48

Operating Notes for the VLAN Configuration Enhancement . . . 4-50
How to Save Your Current Configuration 4-51

Displaying MSTP Statistics and Configuration 4-53
Displaying Global MSTP Status 4-54
Displaying Detailed Port Information 4-56
Displaying Status for a Specific MST Instance 4-57
Displaying the MSTP Configuration 4-58

Troubleshooting an MSTP Configuration 4-62
Displaying the Change History of Root Bridges 4-62
Displaying Debug Counters for All MST Instances 4-65
Displaying Debug Counters for One MST Instance 4-66
Displaying Debug Counters for Ports in an MST Instance 4-68
Field Descriptions in MSTP Debug Command Output 4-70
Troubleshooting MSTP Operation 4-73

Loop Protection 4-74
Configuring Loop Protection 4-75
Viewing Loop Protection Status 4-76

Overview

The switches covered in this guide, use the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard.

MSTP Features

802.1s Spanning Tree Protocol	Default Setting	Page Ref
Viewing MSTP Status and Configuration	n/a	page 4-53
Configuring MSTP Operation Mode and Global Parameters	Disabled	page 4-21 and following
Configuring Basic Port Connectivity Parameters	admin-edge-port: No-disabled auto-edge-port: Yes-enabled bpdu-filter: No-disabled bpdu-protection: No-disabled hello-time: 2 path-cost: auto point-to-point MAC: Force-True priority: 128 (multiplier: 8) root-guard: No-disabled tcn-guard: No-disabled loop protection: Send disable	page 4-27 and following
Configuring MSTP Instance Parameters	instance (MSTPI): none priority: 32768 (multiplier: 8)	page 4-39
Configuring MSTP Instance Per-Port Parameters	path-cost: auto priority: 128 (multiplier: 8)	page 4-41
Enabling/Disabling MSTP Spanning Tree Operation	Disabled	page 4-44
Enabling an Entire MST Region at Once	n/a	page 4-44

Without spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages, leading to a "broadcast storm" that can bring down the network.

Note

MSTP cannot protect against loops when there is an unmanaged device on the network that drops spanning tree packets, or may fail to detect loops where this is an edge port configured with client authentication (802.1X, Web and MAC authentication). To protect against the formation of loops in these cases, you can use the loop protection feature (see "Loop Protection" on page 4-74).

Multiple-Instance spanning tree operation (802.1s) ensures that only one active path exists between any two nodes in a spanning-tree *instance*. A spanning-tree instance comprises a unique set of VLANs, and belongs to a specific spanning-tree *region*. A region can comprise multiple spanning-tree instances (each with a different set of VLANs), and allows one active path among regions in a network. Applying VLAN tagging to the ports in a multiple-instance spanning-tree network enables blocking of redundant links in one instance while allowing forwarding over the same links for non-redundant use by another instance.

For example, suppose you have three switches in a region configured with VLANs grouped into two instances, as follows:

VLANs	Instance 1	Instance 2
10, 11, 12	Yes	No
20, 21, 22	No	Yes

The logical and physical topologies resulting from these VLAN/Instance groupings result in blocking on different links for different VLANs:

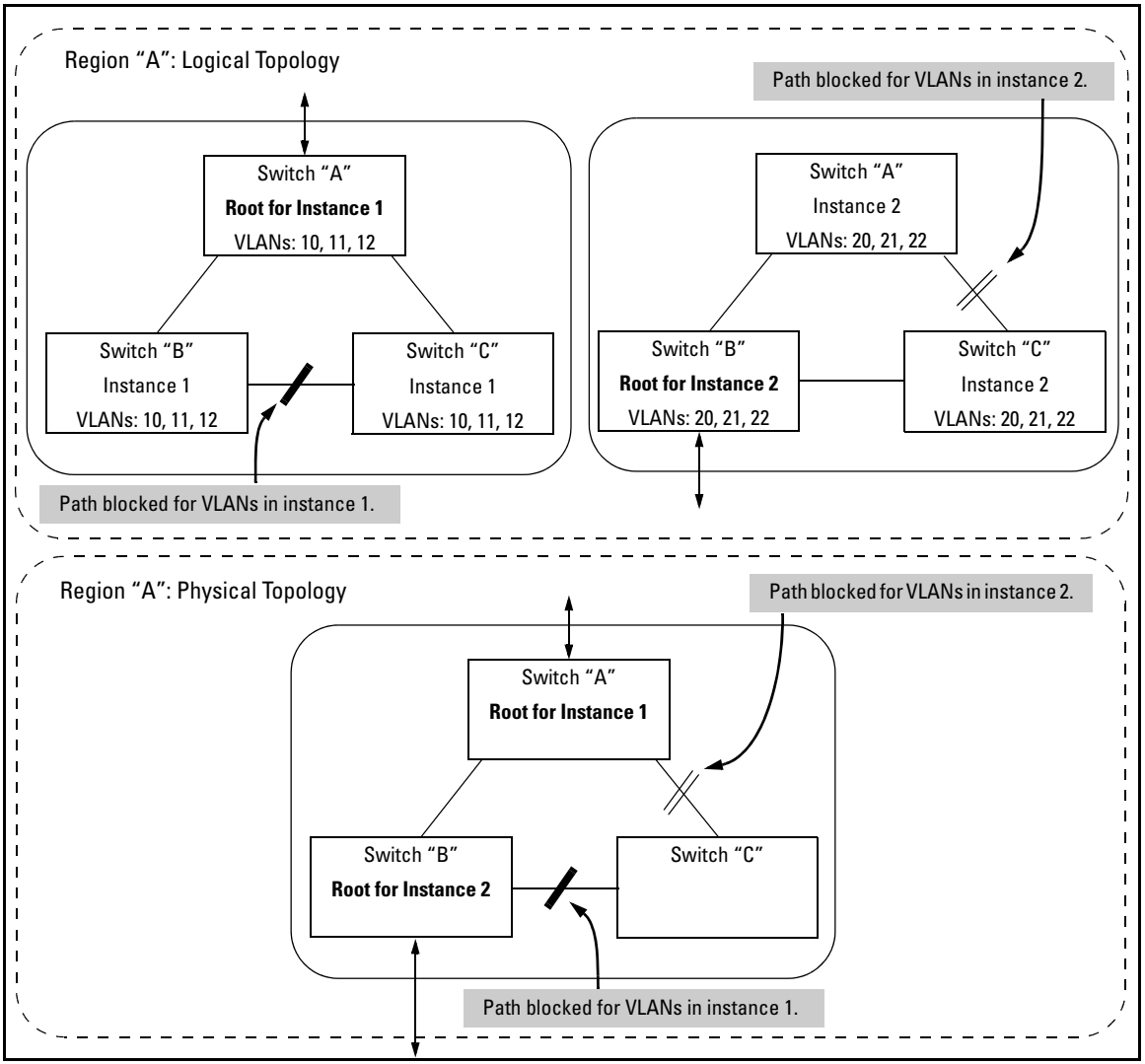


Figure 4-1. Example of a Multiple Spanning-Tree Application

802.1s Multiple Spanning Tree Protocol (MSTP)

The 802.1D and 802.1w spanning tree protocols operate without regard to a network's VLAN configuration, and maintain one common spanning tree throughout a bridged network. Thus, these protocols map one loop-free, logical topology on a given physical topology. The 802.1s Multiple Spanning Tree protocol (MSTP) uses VLANs to create multiple spanning trees in a network, which significantly improves network resource utilization while maintaining a loop-free environment.

While the per-VLAN spanning tree approach adopted by some vendors overcomes the network utilization problems inherent in using STP or RSTP, using a per-VLAN technology with multiple VLANs can overload the switch's CPU. MSTP on the switches covered in this guide complies with the IEEE 802.1s standard, and extends STP and RSTP functionality to map multiple independent spanning tree instances onto a physical topology. With MSTP, each spanning tree instance can include one or more VLANs and applies a separate, per-instance forwarding topology. Thus, where a port belongs to multiple VLANs, it may be dynamically blocked in one spanning tree instance, but forwarding in another instance. This achieves load-balancing across the network while keeping the switch's CPU load at a moderate level (by aggregating multiple VLANs in a single spanning tree instance). MSTP provides fault tolerance through rapid, automatic reconfiguration if there is a failure in a network's physical topology.

With MSTP-capable switches, you can create a number of MST regions containing multiple spanning tree instances. This requires the configuration of a number of MSTP-capable switches. However, it is NOT necessary to do this. You can just enable MSTP on an MSTP-capable switch and a spanning tree instance is created automatically. This instance always exists by default when spanning tree is enabled, and is the spanning tree instance that communicates with STP and RSTP environments. The MSTP configuration commands operate exactly like RSTP commands and MSTP is backward-compatible with the RSTP-enabled and STP-enabled switches in your network.

Caution

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default MSTP parameter settings are usually adequate for spanning tree operation. Also, because incorrect MSTP settings can adversely affect network performance, you should not change the MSTP settings from their default values unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default MSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the MSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For MSTP information beyond what is provided in this manual, refer to the IEEE 802.1s standard.

MSTP Structure

MSTP maps active, separate paths through separate spanning tree instances and between MST regions. Each MST region comprises one or more MSTP switches. Note that MSTP recognizes an STP or RSTP LAN as a distinct spanning-tree region.

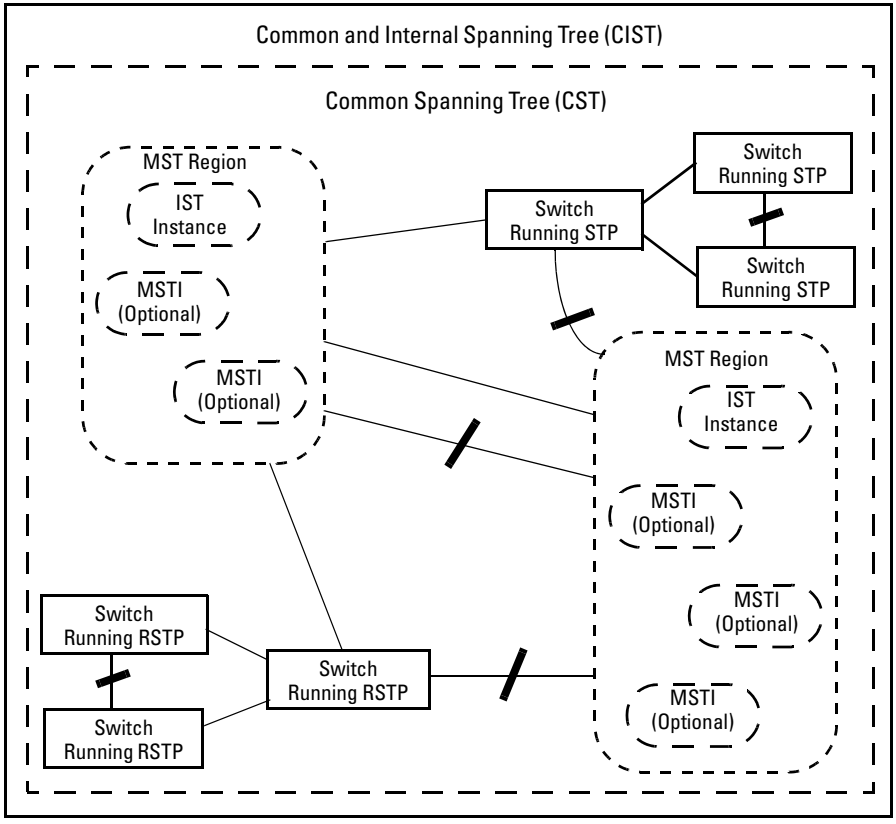


Figure 4-2. Example of MSTP Network with Legacy STP and RSTP Devices Connected

Common and Internal Spanning Tree (CIST): The CIST identifies the regions in a network and administers the CIST root bridge for the network, the root bridge for each region, and the root bridge for each spanning-tree instance in each region.

Common Spanning Tree (CST): The CST administers the connectivity among the MST regions, STP LANs, and RSTP LANs in a bridged network.

MST Region: An MST region comprises the VLANs configured on physically connected MSTP switches. All switches in a given region must be configured with the same VLANs, the same Multiple Spanning Tree Instances (MSTIs), and the same MST configuration identifiers.

Internal Spanning Tree (IST): The IST administers the topology within a given MST region. When you configure a switch for MSTP operation, the switch automatically includes all of the static VLANs configured on the switch in a single, active spanning tree topology (instance) within the IST. This is termed the “IST instance”. Any VLANs you subsequently configure on the switch are added to this IST instance. To create separate forwarding paths within a region, group specific VLANs into different Multiple Spanning Tree Instances (MSTIs). (Refer to “Multiple Spanning Tree Instance (MSTI)”, below.)

Types of Multiple Spanning Tree Instances: A multiple spanning tree network comprises separate spanning-tree instances existing in an MST region. (There can be multiple regions in a network.) Each instance defines a single forwarding topology for an exclusive set of VLANs. By contrast, an STP or RSTP network has only one spanning tree instance for the entire network, and includes all VLANs in the network. (An STP or RSTP network operates as a single-instance network.) A region can include two types of STP instances:

- **Internal Spanning-Tree Instance (IST Instance):** This is the default spanning tree instance in any MST region. It provides the root switch for the region and comprises all VLANs configured on the switches in the region that are not specifically assigned to Multiple Spanning Tree Instances (MSTIs, described below).

  Within a region, the IST instance provides a loop-free forwarding path for all VLANs associated with it. VLANs that are not associated with an MSTI are, by default, associated with the IST instance. Note that the switch automatically places dynamic VLANs (resulting from GVRP operation) in the IST instance. Dynamic VLANs cannot exist in an MSTI (described below).

- **Multiple Spanning Tree Instance (MSTI):** This type of configurable spanning tree instance comprises all static VLANs you specifically assign to it, and must include at least one VLAN. The VLAN(s) you assign to an

MSTI must initially exist in the IST instance of the same MST region. When you assign a static VLAN to an MSTI, the switch removes the VLAN from the IST instance. (Thus, you can assign a VLAN to only one MSTI in a given region.) All VLANs in an MSTI operate as part of the same single spanning tree topology. (The switch does not allow dynamic VLANs in an MSTI.)

Caution

When you enable MSTP on the switch, the default MSTP spanning tree configuration settings comply with the values recommended in the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard. Note that inappropriate changes to these settings can result in severely degraded network performance. For this reason, *ProCurve strongly recommends that changing these default settings be reserved only for experienced network administrators who have a strong understanding of the IEEE 802.1D/w/s standards and operation.*

How MSTP Operates

In the factory default configuration, spanning tree operation is off. Also, the switch retains its currently configured spanning tree parameter settings when disabled. Thus, if you disable spanning tree, then later re-enable it, the parameter settings will be the same as before spanning tree was disabled. The switch also includes a "Pending" feature that enables you to exchange MSTP configurations with a single command. (Refer to "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another" on page 4-44.)

Note

The switch automatically senses port identity and type, and automatically defines spanning-tree parameters for each type, as well as parameters that apply across the switch. Although these parameters can be adjusted, *ProCurve strongly recommends leaving these settings in their default configurations unless the proposed changes have been supplied by an experienced network administrator who has a strong understanding of the IEEE 802.1D/w/s standards and operation.*

MST Regions

All MSTP switches in a given region must be configured with the same VLANs. Also, each MSTP switch within the same region must have the same VLAN-to-instance assignments. (A VLAN can belong to only one instance within any region.) Within a region:

- All of the VLANs belonging to a given instance compose a single, active spanning-tree topology for that instance.
- Each instance operates independently of other regions.

Between regions there is a single, active spanning-tree topology.

How Separate Instances Affect MSTP Operation. Assigning different groups of VLANs to different instances ensures that those VLAN groups use independent forwarding paths. For example, in figure 4-3 each instance has a different forwarding path.

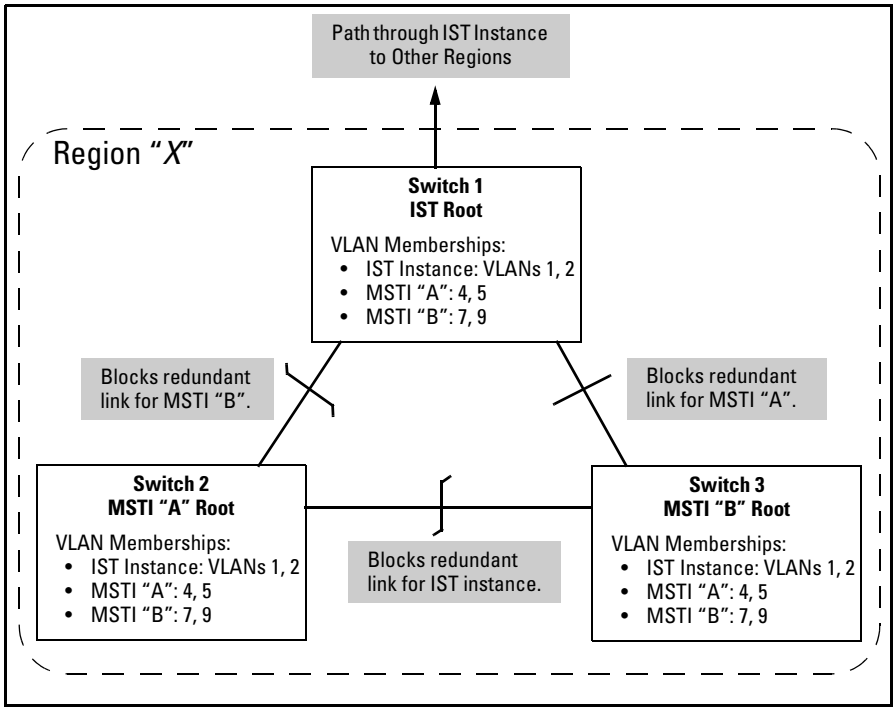

Path through IST Instance to Other Regions
Region "X"
Switch 1
IST Root
VLAN Memberships:
• IST Instance: VLANs 1, 2
• MSTI "A": 4, 5
• MSTI "B": 7, 9
Blocks redundant link for MSTI "B".
Blocks redundant link for MSTI "A".
Switch 2
MSTI "A" Root
VLAN Memberships:
• IST Instance: VLANs 1, 2
• MSTI "A": 4, 5
• MSTI "B": 7, 9
Blocks redundant link for IST instance.
Switch 3
MSTI "B" Root
VLAN Memberships:
• IST Instance: VLANs 1, 2
• MSTI "A": 4, 5
• MSTI "B": 7, 9

Figure 4-3. Active Topologies Built by Three Independent MST Instances

While allowing only one active path through a given instance, MSTP retains any redundant physical paths in the instance to serve as backups (blocked) paths in case the existing active path fails. Thus, if an active path in an instance fails, MSTP automatically activates (unblocks) an available backup to serve as the new active path through the instance for as long as the original active path is down. Note also that a given port may simultaneously operate in different states (forwarding or blocking) for different spanning-tree instances within the same region. This depends on the VLAN memberships to which the port is assigned. For example, if a port belongs to VLAN 1 in the IST instance of a region and also belongs to VLAN 4 in MSTI "*x*" in the same region, the port may apply different states to traffic for these two different instances.

Within a region, traffic routed between VLANs in separate instances can take only one physical path. To ensure that traffic in all VLANs within a region can travel between regions, all of the boundary ports for each region should belong to all VLANs configured in the region. Otherwise, traffic from some areas within a region could be blocked from moving to other regions.

All MSTP switches (as well as STP and RSTP switches) in a network use BPDUs (Bridge Protocol Data Units) to exchange information from which to build multiple, active topologies in the individual instances within a region and between regions. From this information:

- The MSTP switches in each LAN segment determine a designated bridge and designated port or trunk for the segment.
- The MSTP switches belonging to a particular instance determine the root bridge and root port or trunk for the instance.
- For the IST instance within a region, the MSTP switches linking that region to other regions (or to STP or RSTP switches) determine the IST root bridge and IST root port or trunk for the region. (For any Multiple Spanning-Tree instance—MSTI—in a region, the regional root may be a different switch that is not necessarily connected to another region.)
- The MSTP switches block redundant links within each LAN segment, across all instances, and between regions, to prevent any traffic loops.

As a result, each individual instance (spanning tree) within a region determines its regional root bridge, designated bridges, and designated ports or trunks.

Regions, Legacy STP and RSTP Switches, and the Common Spanning Tree (CST)

The IST instance and any MST instances in a region exist only within that region. Where a link crosses a boundary between regions (or between a region and a legacy STP or RSTP switch), traffic is forwarded or blocked as determined by the Common Spanning Tree (CST). The CST ensures that there is only one active path between any two regions, or between a region and a switch running STP and RSTP. (Refer to figure 4-2 on page 4-7.)

MSTP Operation with 802.1Q VLANs

As indicated in the preceding sections, within a given MST instance, a single spanning tree is configured for all VLANs included in that instance. This means that if redundant physical links exist in separate VLANs within the same instance, MSTP blocks all but one of those links. However, you can prevent the bandwidth loss caused by blocked redundant links for different VLANs in an instance by using a port trunk. The following example shows how you can use a port trunk with 802.1Q (tagged) VLANs and MSTP without unnecessarily blocking any links or losing any bandwidth.

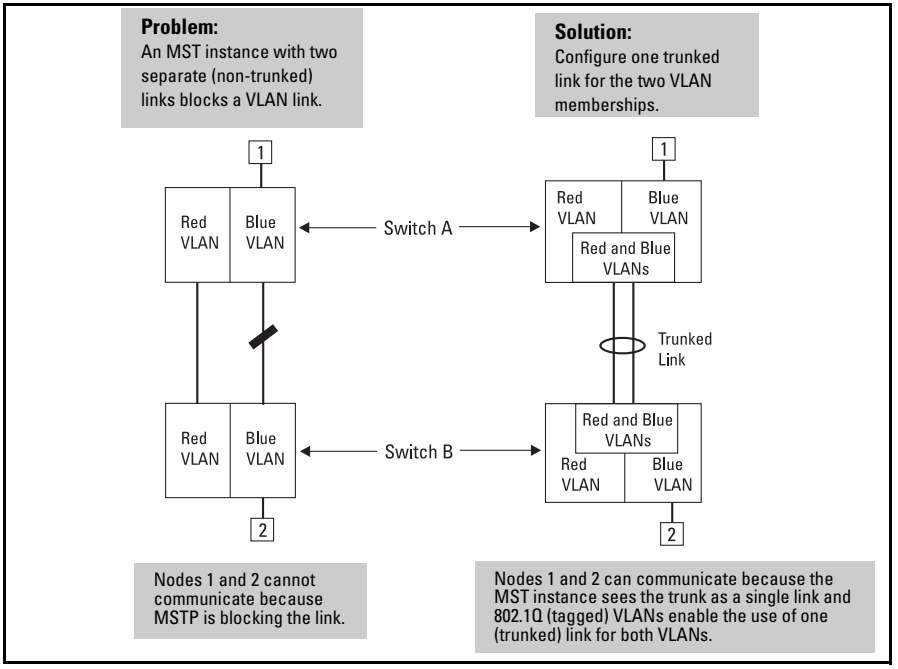

Problem:
An MST instance with two separate (non-trunked) links blocks a VLAN link.
Solution:
Configure one trunked link for the two VLAN memberships.
1
Red VLAN
Blue VLAN
Switch A
Red VLAN
Blue VLAN
Red and Blue VLANs
Trunked Link
Red VLAN
Blue VLAN
Switch B
Red and Blue VLANs
Red VLAN
Blue VLAN
2
Nodes 1 and 2 cannot communicate because MSTP is blocking the link.
Nodes 1 and 2 can communicate because the MST instance sees the trunk as a single link and 802.1Q (tagged) VLANs enable the use of one (trunked) link for both VLANs.

Figure 4-4. Example of Using a Trunked Link To Support Multiple VLAN Connectivity within the Same MST Instance

Note

All switches in a region should be configured with the VLANs used in that region, and all ports linking MSTP switches together should be members of all VLANs in the region. Otherwise, the path to the root for a given VLAN will be broken if MSTP selects a spanning tree through a link that does not include that VLAN.

Terminology

BPDU — Acronym for bridge protocol data unit. BPDUs are data messages that are exchanged between the switches within an extended LAN that use a spanning tree protocol topology. BPDU packets contain information on ports, addresses, priorities and costs and ensure that the data ends up where it was intended to go. BPDU messages are exchanged across bridges to detect loops in a network topology. The loops are then removed by placing redundant switch ports in a backup, or blocked, state.

BPDU Filtering — Spanning-tree configuration mode that prevents the switch from receiving and transmitting BPDU frames on a specific port (see page 4-30 for details).

BPDU Protection — Spanning-tree configuration mode which disables a port where BPDU frames are received (see page 4-31 for details).

Bridge: See "MSTP Bridge".

Common and Internal Spanning Tree (CIST): Comprises all LANs, STP, and RSTP bridges and MSTP regions in a network. The CIST automatically determines the MST regions in a network and defines the root bridge (switch) and designated port for each region. The CIST includes the Common Spanning Tree (CST), the Internal Spanning Tree (IST) within each region, and any multiple spanning-tree instances (MSTIs) in a region.

Common Spanning Tree (CST): Refers to the single forwarding path the switch calculates for STP (802.1D) and RSTP (802.1w) topologies, and for inter-regional paths in MSTP (802.1s) topologies. Note that all three types of spanning tree can interoperate in the same network. Also, the MSTP switch interprets a device running 802.1D STP or 802.1w RSTP as a separate region. (Refer to figure 4-2 on page 4-7.)

Internal Spanning Tree (IST): Comprises all VLANs within a region that are not assigned to a multiple spanning-tree instance configured within the region. All MST switches in a region should belong to the IST. In a given region "X", the IST root switch is the regional root switch and provides information on region "X" to other regions.

MSTP (Multiple Spanning Tree Protocol): A network supporting MSTP allows multiple spanning tree instances within configured regions, and a single spanning tree among regions, STP bridges, and RSTP bridges.

MSTP BPDU (MSTP Bridge Protocol Data Unit): These BPDUs carry region-specific information, such as the region identifier (region name and revision number). If a switch receives an MSTP BPDU with a region identifier that differs from its own, then the port on which that BPDU was received is on the boundary of the region in which the switch resides.

MSTP Bridge: In this manual, an MSTP bridge is a switch (or another 802.1s-compatible device) configured for MSTP operation.

MST Region: An MST region forms a multiple spanning tree domain and is a component of a single spanning-tree domain within a network. For switches internal to the MST region:

- All switches have identical MST configuration identifiers (region name and revision number).
- All switches have identical VLAN assignments to the region's IST and (optional) MST instances.
- One switch functions as the designated bridge (IST root) for the region.
- No switch has a point-to-point connection to a bridging device that cannot process RSTP BPDUs.

RSTP — Rapid Spanning Tree Protocol, defined in IEEE 802.1w and ratified in IEEE 802.1D-2004.

Spanning-tree — Generic term to refer to the many spanning-tree flavors: now deprecated STP, RSTP and VLAN-aware MSTP.

STP — Spanning Tree Protocol, part of the original IEEE 802.1D specification. The 2004 edition completely deprecates STP. Both RSTP and MSTP have fallback modes to handle STP.

SNMP — Simple Network Management Protocol, used to remotely manage network devices.

Operating Rules

- All switches in a region must be configured with the same set of VLANs, as well as the same MST configuration name and MST configuration number.
- Within a region, a VLAN can be allocated to either a single MSTI or to the region's IST instance.
- All switches in a region must have the same VID-to-MST instance assignment.
- There is one root MST switch per configured MST instance.
- Because boundary ports provide the VLAN connectivity between regions, all boundary ports on a region's root switch should be configured as members of all static VLANs defined in the region.
- There is one root switch for the Common and Internal Spanning Tree (CIST). At any given time, all switches in the network will use the per-port **hello-time** parameter assignments configured on the CIST root switch.
- Where multiple MST regions exist in a network, there is only one active, physical communication path between any two regions, or between an MST region and an STP or RSTP switch. MSTP blocks any other physical paths as long as the currently active path remains in service.
- Within a network, an MST region appears as a virtual RSTP bridge to other spanning tree entities (other MST regions, and any switches running 802.1D or 802.1w spanning-tree protocols).
- Within an MSTI, there is one physical communication path between any two nodes, regardless of how many VLANs belong to the MSTI. Within an IST instance, there is also one spanning tree across all VLANs belonging to the IST instance.
- An MSTI comprises a unique set of VLANs and forms a single spanning-tree instance within the region to which it belongs.
- A dynamic VLAN learned by GVRP will always be placed in the IST instance and cannot be moved to any configured MST instance.

  Starting in software release 13.*x*.*x*, dynamically learned GVRP VLANs can be mapped to MSTIs and support MSTP load balancing.
- In software release 13.*x*.*x* and later, you can preconfigure static and dynamic VLAN ID-to-MSTI mappings before the VLAN is created on the switch. Later, when the static VLAN ID is configured or a dynamic GVRP VLAN is learned, the VLAN is automatically associated with the preconfigured MSTI. For more information, refer to the **spanning-tree instance vlan** command description on page 4-40.
- Communication between MST regions uses a single spanning tree.

- If a port on a switch configured for MSTP receives a legacy (STP/802.1D or RSTP/802.1w) BPDU, it automatically operates as a legacy port. In this case, the MSTP switch interoperates with the connected STP or RSTP switch as a separate MST region.
- Within an MST region, there is one logical forwarding topology per instance, and each instance comprises a unique set of VLANs. Where multiple paths exist between a pair of nodes using VLANs belonging to the same instance, all but one of those paths will be blocked for that instance. However, if there are different paths in different instances, all such paths are available for traffic. Separate forwarding paths exist through separate spanning tree instances.
- A port can have different states (forwarding or blocking) for different instances (which represent different forwarding paths).
- MSTP interprets a switch mesh as a single link.

MSTP Compatibility with RSTP or STP

IEEE 802.1s MSTP includes RSTP functionality and is designed to be compatible with both IEEE 802.1D and 802.1w spanning-tree protocols. Using the default configuration values, your switches will interoperate effectively with RSTP and STP devices. MSTP automatically detects when the switch ports are connected to non-MSTP devices in the spanning tree and communicates with those devices using 802.1D or 802.1w STP BPDU packets, as appropriate.

To enable effective interoperation with STP (802.1D) configured devices, however, you may need to adjust the default configuration values. Here are two such examples:

- The rapid state transitions employed by MSTP may result in an increase in the rates of frame duplication and misordering in the switched LAN. To allow the switch to support applications and protocols that may be sensitive to frame duplication and misordering, you can disable rapid transitions by setting the Force Protocol Version parameter to **STP-compatible**. The value of this parameter applies to all ports on the switch. See information on **force version** on page 4-23.
- One of the benefits of MSTP is the implementation of a larger range of port path costs, which accommodates higher network speeds. However, this can create some incompatibility between devices running the older 802.1D STP. You can adjust to this incompatibility by implementing the global **spanning-tree legacy-path cost** command (see page 4-23). See also the “Note on Path Cost” below.

Note on Path Cost

RSTP and MSTP implement a greater range of path costs than 802.1D STP, and use different default path cost values to account for higher network speeds. These values are shown below.

Port Type	802.1D STP Path Cost	RSTP and MSTP Path Cost
10 Mbps	100	2 000 000
100 Mbps	10	200 000
1 Gbps	5	20 000

Because the maximum value for the path cost allowed by 802.1D STP is 65535, devices running that version of spanning tree cannot be configured to match the values defined by MSTP, at least for 10 Mbps and 100 Mbps ports. In LANs where there is a mix of devices running 802.1D STP, RSTP, and/or MSTP, you should reconfigure the devices so the path costs match for ports with the same network speeds.

Configuring MSTP

This section outlines the main pre-requisites for configuring MSTP in your network, and describes MSTP settings at the global level, per individual port, and per MST instance.

Planning an MSTP Application

Before configuring MSTP, keep in mind the following tips and considerations:

- Ensure that the VLAN configuration in your network supports all of the forwarding paths necessary for the desired connectivity. All ports connecting one switch to another within a region and one switch to another between regions should be configured as members of all VLANs configured in the region.
- Configure all ports or trunks connecting one switch to another within a region as members of all VLANs in the region. Otherwise, some VLANs could be blocked from access to the spanning-tree root for an instance or for the region.

- Plan individual regions based on VLAN groupings. That is, plan on all MSTP switches in a given region supporting the same set of VLANs. Within each region, determine the VLAN membership for each spanning-tree instance. (Each instance represents a single forwarding path for all VLANs in that instance.)
- Verify that there is one logical spanning-tree path through the following:
  - Any inter-regional links
  - Any IST or MST instance within a region
  - Any legacy (802.1D or 802.1w) switch or group of switches. (Where multiple paths exist between an MST region and a legacy switch, expect the CST to block all but one such path.)
- Determine the root bridge and root port for each instance.
- Determine the designated bridge and designated port for each LAN segment.
- Determine which VLANs to assign to each instance, and use port trunks with 802.1Q VLAN tagging where separate links for separate VLANs would result in a blocked link preventing communication between nodes on the same VLAN. (Refer to "MSTP Operation with 802.1Q VLANs" on page 4-12.)
- Identify the edge ports connected to end nodes and enable the admin-edge-port setting for these ports. Leave the admin-edge-port setting disabled for ports connected to another switch, a bridge, or a hub.

Note on MSTP Rapid State Transitions

Under some circumstances the rapid state transitions employed by MSTP can increase the rates of frame duplication and misordering in the switched LAN. To allow MSTP switches to support applications and protocols that may be sensitive to frame duplication and misordering, setting the Force Protocol Version (**force-version**) parameter to **stp-compatible** allows MSTP to operate with rapid transitions disabled. The value of this parameter applies to all ports on the switch. See the information on **force-version** on page 4-23.

MSTP Configuration Overview

This section describes the general steps for configuring MSTP via the CLI, assuming that you have already determined the VLANs you want MSTP to use (see "Planning an MSTP Application" on page 4-17). A description of each MSTP command syntax is provided in the following sections.

1. **Configure MSTP global parameters**.

   This step involves configuring the following:

   - Required parameters for MST region identity:

     Region Name: **spanning-tree config-name**

     Region Revision Number: **spanning-tree config-revision**

   - Optional MSTP parameter changes for region settings:

     ProCurve recommends that you leave these parameters at their default settings for most networks. See the "Caution" on page 4-9.

     – The maximum number of hops before the MSTP BPDU is discarded: **spanning-tree max-hops** (default: 20)

     – Force-Version operation: **spanning-tree force-version**

     – Forward Delay: **spanning-tree forward-delay**

     – Hello Time (if it is the root device): **spanning-tree hello-time**

     – Maximum age to allow for STP packets before discarding: **spanning-tree maximum-age**

     – Device spanning-tree priority. Specifies the priority value used along with the switch MAC address to determine which device is root. The lower a priority value, the higher the priority. **spanning-tree priority**

2. **Configure per port parameters.**

   ProCurve recommends that you use the default settings for these parameters and apply changes on a per-port basis only where a non-default setting is clearly indicated by the circumstances of individual links. Other features you might consider include BPDU Filtering or BPDU Protection—these provide additional per-port control over spanning-tree operations and security on the switch.

3. **Configure MST instances.**
   - Configure one instance for each VLAN group that you want to operate as an active topology within the region to which the switch belongs. When you create the instance, you must include a minimum of one VID. You can add more VIDs later if desired.

     spanning-tree instance < *n* > **vlan** < *vid* >

     To move a VLAN from one instance to another, first use **no spanning-tree instance** < *n* > **vlan** < *vid* > to unmap the VLAN from the current instance, then add the VLAN to the other instance. (While the VLAN is unmapped from an MSTI, it is associated with the region's IST instance.)

4. **Configure the priority for each instance.**
   spanning-tree instance < *n* > priority < *n* >

5. **Configure MST instance port parameters.**

   ProCurve recommends that you apply changes on a per-port basis only where a non-default setting is clearly indicated by the circumstances of individual links. For example, you might want to set the path cost value for the port(s) used by a specific MST instance.
   spanning-tree instance < *1..16* >< *port-list* > **path-cost** < *auto* | *1..200000000* >

   Alternatively, leaving this setting at the default (auto) allows the switch to calculate the path-cost from the link speed.

6. **Enable spanning-tree operation on the switch**.
   spanning-tree

For an example of a complete MSTP network configuration:

- Go to the ProCurve Web site at **www.procurve.com/support**.
- Click on **Configuration examples**.
- Click on **ProCurve Switch 8212zl**.
- Under **STP Example**, click on **ProCurve & Cisco Spanning Tree Interoperability**.

Configuring MSTP Operation Mode and Global Settings

The commands in this section apply at the switch (global) level. For details of how to configure spanning tree settings on individual ports, see "Configuring MSTP Per-Port Parameters" on page 4-26.

MSTP Global Command	Page
spanning-tree	*
clear-debug-counters	4-21
config-name < *ascii-string* >	4-21
config-revision < *revision-number* >	4-22
force-version < stp-compatible | rstp-operation | mstp-operation>	4-23
forward-delay	4-23
hello-time < 1..10 >	4-24
legacy-mode	4-23
legacy-path-cost	4-23
max-hops < *hop-count* >	4-24
maximum-age	4-24
pending	4-24
priority	4-25
trap errant-bpdu	4-25

** Enabling MSTP operation using the **spanning-tree** global command is the final step in the configuration process. See "Enabling or Disabling Spanning Tree Operation" on page 4-44.*

Syntax: spanning-tree clear-debug-counters

Clears spanning tree debug counters.

Syntax: [no] spanning-tree config-name < *ascii-string* >

This command resets the configuration name of the MST region in which the switch resides. This name can include up to 32 nonblank characters and is case-sensitive. On all switches within a given MST region, the configuration names must be identical. Thus, if you want more than one MSTP switch in the same MST region, you must configure the identical region name on all such switches. If you retain the default configuration name on a switch, it cannot exist in the same MST region with another switch.
(Default Name: A text string using the hexadecimal representation of the switch's MAC address)

The **no** *form of the command overwrites the currently configured name with the default name.*

Note: *This option is available only when the switch is configured for MSTP operation. Also, there is no defined limit on the number of regions you can configure.*

Syntax: spanning-tree config-revision < *revision-number* >

This command configures the revision number you designate for the MST region in which you want the switch to reside. This setting must be the same for all switches residing in the same region. Use this setting to differentiate between region configurations in situations such as the following:

- *Changing configuration settings within a region where you want to track the configuration versions you use*
- *Creating a new region from a subset of switches in a current region and want to maintain the same region name.*
- *Using the* **pending** *option to maintain two different configuration options for the same physical region.*

Note that this setting must be the same for all MSTP switches in the same MST region. (Range: **0 - 65535***; Default:* **0***)*

Note: *This option is available only when the switch is configured for MSTP operation.*

Syntax: spanning-tree force-version < stp-compatible | rstp-operation | mstp-operation >

Sets the spanning-tree compatibility mode. This command forces the switch to emulate behavior of earlier versions of spanning tree protocol, or return to MSTP behavior. The command is useful in test or debug applications, and removes the need to reconfigure the switch for temporary changes in spanning-tree operation.

stp-compatible: *The switch applies 802.1D STP operation on all ports.*

rstp-operation: *The switch applies 802.1w operation on all ports except those ports where it detects a system using 802.1D Spanning Tree.*

mstp-operation: *The switch applies 802.1s MSTP operation on all ports where compatibility with 802.1D or 802.1w spanning tree protocols is not required.*

Note that even when mstp-operation is selected, if the switch detects an 802.1D BPDU or an 802.1w BPDU on a port, it communicates with the device linked to that port using STP or RSTP BPDU packets. Also, if errors are encountered as described in the "Note on MSTP Rapid State Transitions" on page 4-18, setting **force-version** *to* **stp-compatible** *forces the MSTP switch to communicate out all ports using operations that are compatible with IEEE 802.1D STP.*

Syntax: spanning-tree forward-delay

Sets time the switch waits between transitioning from listening to learning and from learning to forwarding states.

(*Range: 4 - 30; Default:* **15**.)

Syntax: spanning-tree legacy-mode

Sets spanning-tree protocol to operate in 802.1D legacy mode (STP-compatible).

(*Default:* **MSTP-operation**.)

The **no** *form of the command returns the switch to the default 802.1s native mode (MSTP-operation).*

Syntax: spanning-tree legacy-path-cost

Sets spanning-tree to operate with 802.1d (legacy) path cost values.

(*Default:* **802.1t**.)

The **no** *form of the command returns the switch to the default 802.1t (not legacy) path cost values.*

Syntax: spanning-tree hello-time < 1..10 >

If MSTP is running and the switch is operating as the CIST root for your network, this command specifies the time in seconds between transmissions of BPDUs for all ports on the switch configured with the **Global** *option. (the default). This parameter applies in MSTP, RSTP and STP modes. During MSTP operation, you can override this global setting on a per-port basis with this command:* **spanning-tree < *port-list* > hello-time < 1..10 >** (see page 4-27). (*Default:* **2**.)

Syntax: spanning-tree max-hops < *hop-count* >

This command resets the number of hops allowed for BPDUs in an MST region. When an MSTP switch receives a BPDU, it decrements the hop-count setting the BPDU carries. If the hop-count reaches zero, the receiving switch drops the BPDU. Note that the switch does not change the message-age and maximum-age data carried in the BPDU as it moves through the MST region and is propagated to other regions. (Range: 1 - 40; Default: 20)

Syntax: spanning-tree maximum age

Sets the maximum age of received STP information before it is discarded.

(*Default:* **20**.)

Syntax: spanning-tree pending < apply | config-name | config-revision | instance | reset >

Manipulates the pending MSTP configuration. The command is useful in test or debug applications, and enables rapid reconfiguration of the switch for changes in spanning-tree operation.

apply: *Apply pending MSTP configuration (swaps active and pending configurations).*

config-name: *Sets the pending MST region configuration name (default is switch's MAC address).*

config-revision: *Sets the pending MST region configuration revision number (default is 0).*

instance: *Change pending MST instance configuration.*

reset: *Copy active configuration to pending.*

Syntax: spanning-tree priority < priority-multiplier >

Every switch running an instance of MSTP has a Bridge Identifier, which is a unique identifier that helps distinguish this switch from all others. The switch with the lowest Bridge Identifier is elected as the root for the tree.

The Bridge Identifier is composed of a configurable Priority component (2 bytes) and the bridge's MAC address (6 bytes). The ability to change the Priority component provides flexibility in determining which switch will be the root for the tree, regardless of its MAC address.

This command sets the switch (bridge) priority for the designated region in which the switch resides. The switch compares this priority with the priorities of other switches in the same region to determine the root switch for the region. The lower the priority value, the higher the priority. (If there is only one switch in the region, then that switch is the root switch for the region.) The root bridge in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance. (Traffic in VLANs assigned to a numbered STP instance in a given region moves to other regions through the root switch for that instance.)

The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) x 4096

*For example, if you configure "***2***" as the priority-multiplier on a given MSTP switch, then the* **Switch Priority** *setting is 8,192.*

Note: *If multiple switches in the same MST region have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that region.*

Syntax: spanning-tree trap errant-bpdu

Enables SNMP traps for errant-BPDUs. Note that this command is designed to be used in conjunction with the spanning-tree bpdu-filter command (see page 4-30) and bpdu-protection command (see page 4-31).

The **no** *form of the command disables traps on the switch.* (*Default:* **Disabled**.)

Configuring MSTP Per-Port Parameters

In an MSTP topology, you configure per-port parameters in the global configuration context.

In most cases, ProCurve recommends that you use the default settings for these parameters and apply changes on a per-port basis only where a non-default setting is clearly indicated by the circumstances of individual links. Some port parameters (such as **admin-edge-port**) affect all MSTI instances that consist of VLANs configured on the port; other port parameters (such as **path-cost**) affect only the specified MST.

Per Port Command	Page
spanning-tree < *port-list* >	
admin-edge-port	below
auto-edge-port	4-27
bpdu-filter	4-30
bpdu-protection	4-32
hello-time < global | 1..10 >	4-27
mcheck	4-28
path-cost < auto | 200000000 >	4-40
point-to-point-mac < force-true | force-false | auto>	4-25
priority <priority-multiplier>	4-25
pvst-filter	4-36
pvst-protection	4-34
root-guard	4-29
tcn-guard	4-30

Configuring Per Port Parameters

Syntax: [no] spanning-tree <port-list> admin-edge-port

Enable **admin-edge-port** *on ports connected to end nodes. During spanning tree establishment, ports with* **admin-edge-port** *enabled transition immediately to the forwarding state. If a bridge or switch is detected on the segment, the port automatically operates as non-edge, not enabled. (Default:* **No** *- disabled)*

If **admin-edge-port** *is disabled on a port and* **auto-edge-port** *has not been disabled, the* **auto-edge-port** *setting controls the behavior of the port.*

The **no spanning-tree < *port-list* > admin-edge-port** *command disables edge-port operation on the specified ports.*

Syntax: [no] spanning-tree < *port-list* > auto-edge-port

Supports the automatic identification of edge ports. The port will look for BPDUs for 3 seconds; if there are none it begins forwarding packets. If **admin-edge-port** *is enabled for a port, the setting for* **auto-edge-port** *is ignored whether set to yes or no. If* **admin-edge-port** *is set to* **No**, *and* **auto-edge-port** *has not been disabled (set to* **No***), then the* **auto-edge-port** *setting controls the behavior of the port. (Default:* **Yes - enabled***)*

The **no spanning-tree < *port-list* > auto-edge-port** *command disables* **auto-edge-port** *operation on the specified ports.*

Syntax: spanning-tree < *port-list* > hello-time < global | 1 - 10 >

When the switch is the CIST root, this parameter specifies the interval (in seconds) between periodic BPDU transmissions by the designated ports. This interval also applies to all ports in all switches downstream from each port in the < *port-list*>*. A setting of* **global** *indicates that the ports in* < *port-list*> *on the CIST root are using the value set by the global spanning-tree* **hello-time** *value (page 4-24). When a given switch "X" is not the CIST root, the per-port* **hello-time** *for all active ports on switch "X" is propagated from the CIST root, and is the same as the* **hello-time** *in use on the CIST root port in the currently active path from switch "X" to the CIST root. (That is, when switch "X" is not the CIST root, then the upstream CIST root's port* **hello-time** *setting overrides the* **hello-time** *setting configured on switch "X").*

(Default Per-Port setting: **Use Global***.*
Default Global Hello-Time: **2***.)*

Syntax: spanning-tree < *port-list* > mcheck

Forces a port to send RST/MST BPDUs for 3 seconds. This tests whether all STP bridges on the attached LAN have been removed and the port can migrate to native MSTP mode and use RST/MST BPDUs for transmission.

Syntax: spanning-tree < *port-list* > path-cost < auto | 1..200000000 >

Assigns an individual port cost that the switch uses to determine which ports are forwarding ports in a given spanning tree. In the default configuration (auto) the switch determines a port's path cost by the port's type:

- *10 Mbps:* **2000000**
- *100 Mbps:* **200000**
- *1 Gbps:* **20000**

Refer to "Note on Path Cost" on page 4-17 for information on compatibility with devices running 802.1D STP for the path cost values

(Default: Auto).

Syntax: spanning-tree < *port-list* > point-to-point-mac < true | false | auto >

This parameter informs the switch of the type of device to which a specific port connects.

true ***(default)****: Indicates a point-to-point link to a device such as a switch, bridge, or end-node.*

false*: Indicates a connection to a hub (which is a shared LAN segment).*

auto: *Causes the switch to set Force-False on the port if it is not running at full duplex. (Connections to hubs are half-duplex.)*

Syntax: spanning-tree < *port-list* > priority < priority-multiplier >

MSTP uses this parameter to determine the port(s) to use for forwarding. The port with the lowest priority number has the highest priority for use. The range is 0 to 240, and is configured by specifying a multiplier from 0 - 15. When you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) *x* 16

*For example, if you configure "***2***" as the priority multiplier on a given port, then the actual* **Priority** *setting is 32. Thus, after you specify the port priority multiplier, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree** *or* **show spanning-tree < *port-list* >** *displays.*

You can view the actual multiplier setting for ports by executing **show running** *and looking for an entry in this format:*

spanning-tree < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "3" results in this line in the **show running** *output:*

```
spanning-tree A2 priority 3
```

Syntax: spanning-tree < *port-list* > root-guard

MSTP only. When a port is enabled as **root-guard**, *it cannot be selected as the root port even if it receives superior STP BPDUs. The port is assigned an "alternate" port role and enters a blocking state if it receives superior STP BPDUs. (A superior BPDU contains "better" information on the root bridge and/or path cost to the root bridge, which would normally replace the current root bridge selection.)*

The superior BPDUs received on a port enabled as **root-guard** *are ignored. All other BPDUs are accepted and the external devices may belong to the spanning tree as long as they do not claim to be the Root device.*

Use this command on MSTP switch ports that are connected to devices located in other administrative network domains to:

- *Ensure the stability of the core MSTP network topology so that undesired or damaging influences external to the network do not enter.*
- *Protect the configuration of the CIST root bridge that serves as the common root for the entire network.*

Default: Disabled.

Syntax: spanning-tree < *port-list* > tcn-guard

When **tcn-guard** *is enabled for a port, it causes the port to stop propagating received topology change notifications and topology changes to other ports.*
(Default: **No** *- disabled)*

Configuring BPDU Filtering

The STP BPDU filter feature allows control of spanning-tree participation on a per-port basis. It can be used to exclude specific ports from becoming part of spanning tree operations. A port with the BPDU filter enabled will ignore incoming BPDU packets and stay locked in the spanning-tree forwarding state. All other ports will maintain their role.

Here are some sample scenarios in which this feature may be used:

- To have STP operations running on selected ports of the switch rather than every port of the switch at a time.
- To prevent the spread of errant BPDU frames.
- To eliminate the need for a topology change when a port's link status changes. For example, ports that connect to servers and workstations can be configured to remain outside of spanning-tree operations.
- To protect the network from denial of service attacks that use spoofing BPDUs by dropping incoming BPDU frames. For this scenario, BPDU protection offers a more secure alternative, implementing port shut down and a detection alert when errant BPDU frames are received (see page 4-32 for details).

Caution

Ports configured with the BPDU filter mode remain active (learning and forward frames); however, spanning-tree cannot receive or transmit BPDUs on the port. The port remains in a forwarding state, permitting all broadcast traffic. This can create a network storm if there are any loops (that is, trunks or redundant links) using these ports. If you suddenly have a high load, disconnect the link and disable the bpdu-filter (using the **no** command).

Command Syntax and Example. The following command is used to configure BPDU filters.

Syntax: [no] spanning-tree <port-list | all> bpdu-filter

Enables/disables the BPDU filter feature on the specified port(s). The bpdu-filter option forces a port to ***always*** *stay in the forwarding state and be excluded from standard STP operation.*

For example, to configure BPDU filtering on port a9, enter:

```
ProCurve(config)# spanning-tree a9 bpdu-filter
```

Viewing BPDU Filtering. The **spanning-tree show** < port> **configuration** command displays the BPDU's filter state.

```
ProCurve(config)# show spanning-tree a9 config

  ...
                  | Path      Prio  Admin Auto  Admin Hello  Root   TCN    BPDU
  Port  Type      | Cost      rity  Edge  Edge  PtP   Time   Guard  Guard  Flt
  ----- --------- + --------- ----- ----- ----- ----- ------ ------ ------ ----
  A9    100/1000T | Auto      128   No    Yes   True  Global No     No     Yes
```

Column showing BPDU filter status

Figure 4-5. Example of BPDU Filter in Show Spanning Tree Configuration Command

BPDU filters per port are displayed as separate entries of the spanning tree category within the configuration file.

```
ProCurve(config)# show configuration
  . . .
  spanning-tree
  spanning-tree A9 bpdu-filter
  spanning-tree C7 bpdu-filter
  spanning-tree Trk2 priority 4
  . . .
```

Rows showing ports with BPDU filters enabled

Figure 4-6. Example of BPDU Filters in the Show Configuration Command

Configuring BPDU Protection

BPDU protection is a security feature designed to protect the active STP topology by preventing spoofed BPDU packets from entering the STP domain. In a typical implementation, BPDU protection would be applied to edge ports connected to end user devices that do not run STP. If STP BPDU packets are received on a protected port, the feature will disable that port and alert the network manager via an SNMP trap as shown in Figure 4-7.

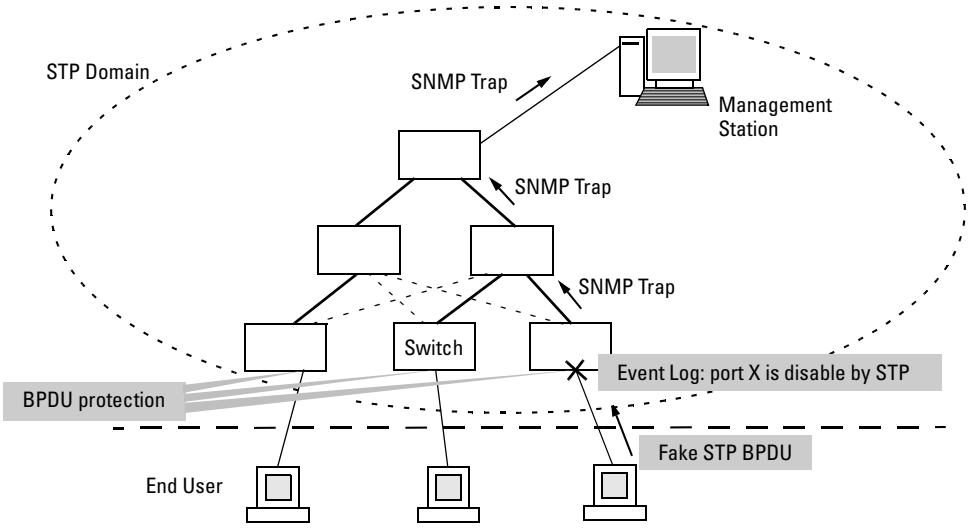


Figure 4-7. Example of BPDU Protection Enabled at the Network Edge

The following commands allow you to configure BPDU protection.

Syntax: [no] spanning-tree <*port-list*> bpdu-protection

Enables/disables the BPDU protection feature on a port

Syntax: [no] spanning-tree <*port-list*> bpdu-protection-timeout <*timeout*>

Configures the duration of time when protected ports receiving unauthorized BPDUs will remain disabled. The default value of 0 (zero) sets an infinite timeout (that is, ports that are disabled by **bpdu-protection** *are not, by default, re-enabled automatically). (Range: 0-65535 seconds; Default: 0)*

Syntax: [no] spanning-tree trap errant-bpdu

Enables/disables the sending of errant BPDU traps.

Caution This command should only be used to guard edge ports that are not expected to participate in STP operations. Once BPDU protection is enabled, it will disable the port as soon as any BPDU packet is received on that interface.

Example. To configure BPDU protection on ports 1 to 10 with SNMP traps enabled, enter:

```
ProCurve(config)# spanning-tree 1-10 bpdu protection
ProCurve(config)# spanning-tree trap errant-bpdu
```

The following steps will then be set in process:

1. When an STP BPDU packet is received on ports 1-10, STP treats it as an unauthorized transmission attempt and shuts down the port that the BPDU came in on.
2. An event message is logged and an SNMP notification trap is generated.
3. The port remains disabled until re-enabled manually by a network administrator using the **interface** <*port-list*> **enable** command.

Note

To re-enable the bpdu-protected ports automatically, configure a timeout period using the **spanning-tree bpdu-protection-timeout** command.

Viewing BPDU Protection Status. The **show spanning-tree bpdu-protection** command displays a summary listing of ports with BPDU protection enabled. To display detailed per port status information, enter the specific port number(s) as shown in Figure 4-8 below.

```
ProCurve(config)# show spanning-tree bpdu-protection a1

 Status and Counters - STP BPDU Protection Information

  BPDU Protection Timeout (sec) : 0
  Protected Ports : A1

  Port Type         Protection    State          Errant BPDUs
  ---- ------------ ------------ -------------- --------------
  A1   100/1000T    Yes          Bpdu Error     1
```

Specifying the port displays additional status information for the designated ports.

Figure 4-8. Example of Show Spanning Tree BPDU Protection Command

BPDU protected ports are displayed as separate entries of the spanning tree category within the configuration file.

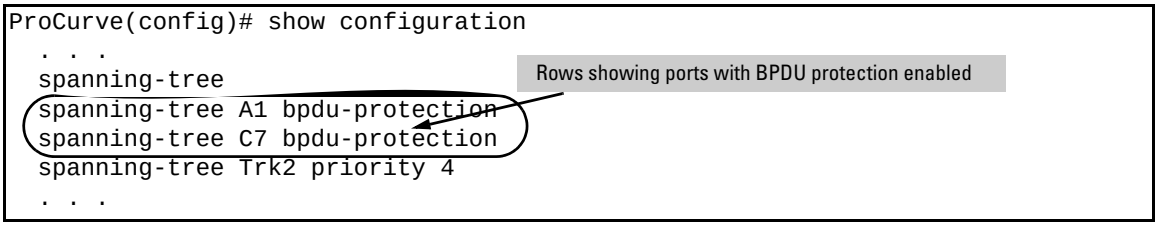


Figure 4-9. Example of BPDU Filters in the Show Configuration Command

PVST Protection and Filtering

Note These options are available for switches that support the MSTP protocol only. They are not supported for switches running RSTP.

PVST Protection

If a ProCurve switch in the core of a network receives Per Vlan Spanning Tree (PVST) BPDUs and forwards the unrecognized PVST BPDUs on to MSTP-only switches, those switches then disconnect themselves from the network. This can create instability in the network infrastructure.

When the PVST protection feature is enabled on a port and a PVST BPDU is received on that port, the interface on which the PVST BPDU arrived is shut down, which isolates the sending switch from the rest of the network. An event message is logged and an SNMP notification trap is generated. The errant BPDU counter hpSwitchStpPortErrantBpduCounter is incremented. The PVST protection feature is enabled per-port.

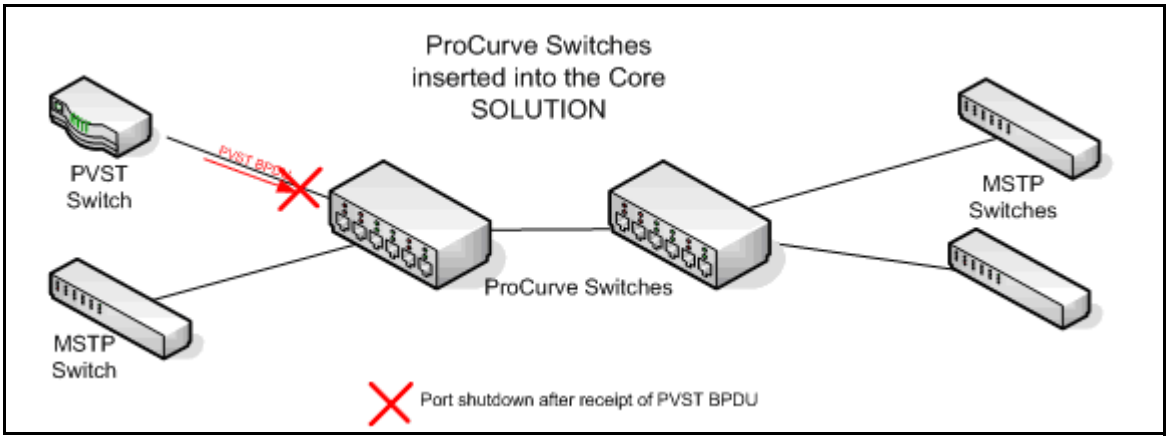


Figure 4-10. PVST Switch Being Isolated after Sending a PVST BPDU

Note

This is similar to the BPDU Guard feature where BPDU protection is applied to edge ports connected to end user devices that do not run STP. If STP BPDU packets are received on a protected port, the feature will disable that port and alert the network manager via an SNMP trap.

Syntax: [no] spanning-tree <port-list> pvst-protection

Enables or disables the PVST protection feature on the port or range of ports specified. The command indicates which ports are not expected to receive any PVST BPDUs.

Default: *Disabled on all ports*

For example, to enable the PVST protection feature on ports 4 through 8, enter this command:

```
ProCurve(config)# spanning-tree 4-8 pvst-protection
```

To disable the PVST protection feature on a port, for example, port 4, use this command:

```
ProCurve(config)# no spanning-tree 4 pvst-protection
```

PVST FilteringI

If you configure a port for PVST filtering instead of PVST protection, the port remains in operation but traps are still generated and the BPDU counter hpSwitchStpPortErrantBpduCounter is incremented.

Caution

Enabling the PVST filter feature allows the port to continuously forward packets without spanning-tree intervention, which could result in loop formation. If this occurs, disable the port and then reconfigure it with these commands:

```
no spanning-tree <port-list> bpdu-filter
no spanning-tree <port-list> pvst-filter
```

Syntax: [no] spanning-tree <port-list> pvst-filter

Enables or disables the PVST filter feature on the port or range of ports specified. The command indicates which ports are not expected to receive any PVST BPDUs.

Default: *Disabled on all ports*

```
ProCurve(config)# spanning-tree 8 pvst-filter

Warning: The BPDU filter allows the port to go into a continuous
         forwarding mode and spanning-tree will not interfere, even if
         the port would cause a loop to form in the network topology.
         If you suddenly experience high traffic load, disable the port
         and reconfigure the BPDU filter with the CLI command(s):
           "no spanning-tree PORT_LIST bpdu-filter"
           "no spanning-tree PORT_LIST pvst-filter"
```

Figure 4-11. Example of Enabling PVST Filtering on a Port

Manually Re-enabling a Port

You can re-enable ports manually or use the automatic re-enable timer command as shown:

```
ProCurve(config)# spanning-tree
                    bpdu-protection-timeout 120
```

Syntax: [no] spanning-tree bpdu-protection-timeout <*timeout*>

Configures the duration of time protected ports remain disabled. The default value of 0 (zero) sets an infinite timeout (that is, ports that are disabled are not, by default, re-enabled automatically).
Note: *This is a GLOBAL command.*
(Range: 0-65535 seconds; Default: 0)

You can also set the timeout in the MIB with this MIB object:

hpSwitchStpBpduProtectionTimeout

Showing Ports Configured with PVST Protection and Filtering

To show which ports are configured for PVST protection, enter this command:

```
ProCurve(config)# show spanning-tree pvst-protection
```

```
ProCurve(config)# show spanning-tree pvst-protection

 Status and Counters - PVST Port(s) BPDU Protection Information

 BPDU Protection Timeout (sec) : 0
 PVST Protected Ports : 5-6
```

Figure 4-12. Example of Show Spanning-tree Command Displaying All Ports with PVST Protection Enabled

To show which ports are configured for PVST filtering, enter this command:

```
ProCurve(config)# show spanning-tree pvst-filter
```

```
ProCurve(config)# show spanning-tree pvst-filter

 Status and Counters - PVST Port(s) BPDU Filter Information

 PVST Filtered Ports : 8
```

Figure 4-13. Example of Show Spanning-tree Command Displaying All Ports with PVST Filtering Enabled

The **show spanning-tree <port-list> detail** command indicates which ports have PVST protection and/or PVST Filtering enabled.

```
.ProCurve(config)# show spanning-tree 7 detail
.
.
.

Port                      : 7
  Status                    : Down
  BPDU Protection           : Yes
  BPDU Filtering            : No
  PVST Protection           : Yes
  PVST Filtering            : No
  Errant BPDU Count         : 0
  Root Guard                : No
  TCN Guard                 : No
  .
  .
  .
```

Figure 4-14. Example of Show Spanning-tree Command Displaying PVST Protection Enabled (Yes)

Configuring MST Instance Parameters

When you enable MSTP on the switch, a spanning tree instance is enabled automatically. The switch supports up to sixteen configurable MST instances for each VLAN group that you want to operate as an active topology within the region to which the switch belongs. When creating an instance, you must include a minimum of one VID. You can add more VIDs later if desired.

Command	Page
[no] spanning-tree instance < 1..16 > vlan < *vid*> [*vid..vid*] no spanning-tree instance < 1..16 >	4-27
spanning-tree instance < 1..16 > priority < 0..15 >	4-39

Syntax: [no] spanning-tree instance < 1..16 > vlan < *vid* [*vid..vid*] >
no spanning-tree instance < 1..16 >

Configuring MSTP on the switch automatically configures the IST instance and places all statically and dynamically configured VLANs on the switch into the IST instance. This command creates a new MST instance (MSTI) and moves the VLANs you specify from the IST to the MSTI.

You must map at least one VLAN to an MSTI when you create it. You cannot map a VLAN ID to more than one instance. You can create up to 16 MSTIs in a region.

The **no** *form of the command removes one or more VLANs from the specified MSTI. If no VLANs are specified, the* **no** *form of the command deletes the specified MSTI.*

When you remove a VLAN from an MSTI, the VLAN returns to the IST instance, where it can remain or be re-assigned to another MSTI configured in the region.

Note: *Starting in software release 13.x.x, you can enter the* ***spanning-tree instance vlan*** *command before a static or dynamic VLAN is configured on the switch to preconfigure VLAN ID-to-MSTI mappings. No error message is displayed. Later, each newly configured VLAN that has already been associated with an MSTI is automatically assigned to the MSTI.*

This new default behavior differs from automatically including configured (static and dynamic) VLANs in the IST instance and requiring you to manually assign individual static VLANs to an MSTI.

Syntax: [no] spanning-tree instance < 1..16 > vlan < *vid* [*vid..vid*] >
no spanning-tree instance < 1..16 >

— Continued —

Note: *The valid VLAN IDs that you can map to a specified MSTI are from 1 to 4094. The VLAN ID-to-MSTI mapping does not require a VLAN to be already configured on the switch. The MSTP VLAN enhancement allows you to preconfigure MSTP topologies before the VLAN IDs associated with each instance exist on a switch.*

When you use preconfigured VLAN ID-to-MSTI topologies, ensure that MSTP switches remain in the same region by mapping all VLAN IDs used in the region to the same MSTIs on each regional switch.

When you upgrade switch software to release 13.x.x and later, the existing MSTP topology configuration is automatically saved. All existing VLAN ID-to-MSTI assignments are maintained on a switch for uninterrupted MSTP network operation.

Syntax: spanning-tree instance < 1..16 > priority < priority-multiplier >

This command sets the switch (bridge) priority for the designated instance. This priority is compared with the priorities of other switches in the same instance to determine the root switch for the instance. The lower the priority value, the higher the priority. (If there is only one switch in the instance, then that switch is the root switch for the instance.) The IST regional root bridge provides the path to instances in other regions that share one or more of the same VLAN(s).

The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch for the specified MST instance is:

(*priority-multiplier*) x 4096

*For example, if you configure "***5***" as the priority-multiplier for MST Instance 1 on a given MSTP switch, then the* **Switch Priority** *setting is 20,480 for that instance in that switch.*

Note: *If multiple switches in the same MST instance have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that instance.*

Configuring MST Instance Per-Port Parameters

Command	Page
spanning-tree instance < 1..16 > < port-list > path-cost < auto | 1..200000000 >	4-41
spanning-tree instance < 1..16 > < *port-list* > priority < *priority-multiplier* >	4-42
spanning-tree < *port-list* > priority < *priority-multiplier* >	4-43

Syntax: spanning-tree instance < 1..16 > < *port-list* > path-cost < auto | 1..200000000 >

This command assigns an individual port cost for the specified MST instance. (For a given port, the path cost setting can be different for different MST instances to which the port may belong.) The switch uses the path cost to determine which ports are the forwarding ports in the instance; that is which links to use for the active topology of the instance and which ports to block. The settings are either **auto** *or in a range from 1 to 200,000,000. With the* **auto** *setting, the switch calculates the path cost from the link speed:*

10 Mbps — 2000000

100 Mbps — 200000

1 Gbps — 20000

(Default: **Auto***)*

Syntax: spanning-tree instance < 1..16 >< *port-list* > priority <*priority-multiplier*>

This command sets the priority for the specified port(s) in the specified MST instance. (For a given port, the priority setting can be different for different MST instances to which the port may belong.) The priority range for a port in a given MST instance is 0-255. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) *x* 16

*For example, if you configure "***2***" as the priority multiplier on a given port in an MST instance, then the actual* **Priority** *setting is 32. Thus, after you specify the port priority multiplier in an instance, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree instance < 1..16 >** *or* **show spanning-tree < *port-list* > instance < 1..16 >** *displays.*

You can view the actual multiplier setting for ports in the specified instance by executing **show running** *and looking for an entry in this format:*

spanning-tree instance < 1..15 > < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "3" in instance 1, results in this line in the **show running** *output:*

```
spanning-tree instance 1 A2 priority 3
```

Syntax: spanning-tree < *port-list* > priority < *priority-multiplier* >

This command sets the priority for the specified port(s) for the IST (that is, Instance 0) of the region in which the switch resides. The "priority" component of the port's "Port Identifier" is set. The Port Identifier is a unique identifier that helps distinguish this switch's ports from all others. It consists of the Priority value with the port number extension—PRIORITY:PORT_NUMBER. A port with a lower value of Port Identifier is more likely to be included in the active topology.

This priority is compared with the priorities of other ports in the IST to determine which port is the root port for the IST instance. The lower the priority value, the higher the priority. The IST root port (or trunk) in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance.

The priority range for a port in a given MST instance is 0-240. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(*priority-multiplier*) x 16

*For example, configuring "***5***" as the priority multiplier on a given port in the IST instance for a region creates an actual* **Priority** *setting of* **80***. Thus, after you specify the port priority multiplier for the IST instance, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree instance ist** *or* **show spanning-tree < *port-list* > instance ist** *displays. You can view the actual multiplier setting for ports in the IST instance by executing* **show running** *and looking for an entry in this format:*

spanning-tree < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "2" in the IST instance, results in this line in the **show running** *output:*

```
spanning-tree A2 priority 2
```

Enabling or Disabling Spanning Tree Operation

This command enables or disables spanning tree operation for any spanning tree protocol enabled on the switch. Before using this command to enable spanning tree, ensure that the version you want to use is active on the switch.

Syntax: [no] spanning-tree

Enabling spanning tree with MSTP configured implements MSTP for all physical ports on the switch, according to the VLAN groupings for the IST instance and any other configured instances. Disabling MSTP removes protection against redundant loops that can significantly slow or halt a network.

This command simply turns spanning tree on or off. It does not change the existing spanning tree configuration.

Note

The convergence time for implementing MSTP changes can be disruptive to your network. To minimize such disruption, consider using the **spanning-tree pending** command (refer to the following section on "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another").

Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another

This operation exchanges the currently active MSTP configuration with the currently pending MSTP configuration. It enables you to implement a new MSTP configuration with minimal network disruption or to exchange MSTP configurations for testing or troubleshooting purposes.

When you configure or reconfigure MSTP, the switch re-calculates the corresponding network paths. This can have a ripple effect throughout your network as adjacent MSTP switches recalculate network paths to support the configuration changes invoked in a single switch. Although MSTP employs rapid spanning-tree operation, the convergence time for implementing MSTP changes can be disruptive to your network. However, by using the **spanning-tree pending** feature, you can set up an MSTP on the switch and then invoke all instances of the new configuration at the same time, instead of one at a time.

Syntax: [no] spanning-tree pending < apply | config-name | config-revision | instance | reset >

This command exchanges the currently active MSTP configuration with the current pending MSTP configuration. Options are as follows:

apply*: Exchanges the currently active MSTP configuration with the pending MSTP configuration.*

config-name*: Specifies the pending MST region name. Must be the same for all MSTP switches in the region. (Default: The switch's MAC address.)*

config-revision: *Specifies the pending MST region configuration revision number. Must be the same for all MSTP switches in the region. (Default: 0).*

instance < *1..16* > **vlan** < *vid | vid-range* >*: Creates the pending instance and assigns one or more VLANs to the instance.*

reset*: Copies the switch's currently active MSTP configuration to the pending configuration. This is useful when you want to experiment with the current MSTP configuration while maintaining an unchanged version.*

To Create a Pending MSTP Configuration. This procedure creates a pending MSTP configuration and exchanges it with the active MSTP configuration:

1. Configure the VLANs you want included in any instances in the new region. When you execute the **pending** command, all VLANs configured on the switch will be assigned to a single pending IST instance unless assigned to other, pending MST instances.(The **pending** command creates the region's IST instance automatically.)
2. Configure MSTP as the spanning-tree protocol, then execute **write mem** and reboot. (The pending option is available only with MSTP enabled.)
3. Configure the pending region **config-name** to assign to the switch.
4. Configure the pending **config-revision** number for the region name.
5. If you want an MST instance other than the IST instance, configure the instance number and assign the appropriate VLANs (VIDs) using the **pending instance** < *1..16* > **vlan** < *vid | vid-range* > command.
6. Repeat step 5 for each additional MST instance you want to configure.

7. To review your pending configuration, use the **show spanning-tree pending** command (see page 4-61).
8. To exchange the currently active MSTP configuration with the pending MSTP configuration, use the **spanning-tree pending apply** command.

MSTP VLAN Configuration Enhancement

Starting in software release 13.*x*.*x*, the MSTP VLAN configuration enhancement allows you to preconfigure an MSTP regional topology and ensure that the same VLAN ID-to-MSTI assignments exist on each MSTP switch in the region.

Caution

When this software version is installed, the prior VLAN ID-to-MSTI mappings do not change. However, this enhancement is not backward-compatible. If you install a software version prior to this version, and you have configured MSTI entries instances mapped to VLANs, they will be removed from the configuration file when booting to the prior version of software. You must do one of the following if you want to install or reload a prior version of the software:

1. Remove all MSTP mappings from the config file and then reconfigure the instance mapping after you are running the desired software version.
2. Save your current configuration file before updating your software to a new version. If you later reload this older version of the software, you can used this configuration file when you reload the older version. See "How to Save Your Current Configuration" on page 4-51.

The default behavior of the **spanning-tree instance vlan** command changes so that, before a static VLAN is configured or a dynamic VLAN is learned on the switch, you can preconfigure its VLAN ID-to-MSTI mapping. Later, when the VLAN is created, it is automatically assigned to the MSTI to which you had previously mapped it.

By supporting preconfigured VLAN ID-to-MSTI topologies, the VLAN Configuration enhancement provides the following benefits:

- Scalability: In a network design in which you plan to use a large number of VLANs, you can preconfigure identical VLAN ID-to-MSTI mappings on all switches in a single, campus-wide MST region, regardless of the specific VLANs that you later configure on each switch. After the initial VLAN ID-to-MSTI mapping, you can decide on the exact VLANs that you need on each switch.

All switches in a region must be configured with the same VLAN ID-to-MSTI mappings and the same MSTP configuration identifiers (region name and revision number).

- Flexibility: By preconfiguring identical VLAN ID-to-MSTI mappings on all switches in an MST region, you can combine switches that support different maximum numbers of VLANs.
- Network stability: You can reduce the interruptions in network connectivity caused by the regeneration of spanning trees in the entire network each time a configuration change in VLAN-to-MSTI mapping is detected on a switch. The negative impact on network performance is reduced if all newly created VLANs are pre-mapped to the correct MST instances. Later, VLAN creation and deletion are ignored by MSTP and no interruption in spanning-tree traffic occurs.
- Usability: Dynamically learned GVRP VLANs can be mapped to MSTIs and support MSTP load balancing.

PreConfiguring VLANs in an MST Instance

When you configure an MSTP regional topology, you create multiple spanning-tree instances. Each MST instance provides a fully connected active topology for a particular set of VLANs.

Each switch in an MSTP region is configured with the following set of common parameters:

- Region name (**spanning-tree config-name**)
- Region revision number (**spanning-tree config-revision**)
- Identical VLAN ID-to-MSTI mapping (**spanning-tree instance vlan**)

Each MST instance supports a different set of VLANs. A VLAN that is mapped to an MST instance cannot be a member of another MST instance.

The MSTP VLAN configuration enhancement allows you to ensure that the same VLAN ID-to-MSTI assignments exist on each MSTP switch in a region. Before a static VLAN is configured or a dynamic VLAN is learned on the switch, you can used the **spanning-tree instance vlan** command to map VLANs to each MST instance in the region. Later, when the VLAN is created, the switch automatically assigns it to the MST instance to which you had previously mapped it.

Syntax: [no] spanning-tree instance < 1..16 > vlan < *vid* [*vid..vid*] >
no spanning-tree instance < 1..16 >

Configuring MSTP on the switch automatically configures the IST instance and places all statically and dynamically configured VLANs on the switch into the IST instance. This command creates a new MST instance (MSTI) and moves the VLANs you specify from the IST to the MSTI.

You must map at least one VLAN to an MSTI when you create it. You cannot map a VLAN ID to more than one instance. You can create up to 16 MSTIs in a region.

The **no** *form of the command removes one or more VLANs from the specified MSTI. If no VLANs are specified, the* **no** *form of the command deletes the specified MSTI.*

When you remove a VLAN from an MSTI, the VLAN returns to the IST instance, where it can remain or be re-assigned to another MSTI configured in the region.

Note: *The valid VLAN IDs that you can map to a specified MSTI are from 1 to 4094. The VLAN ID-to-MSTI mapping does not require a VLAN to be already configured on the switch. The MSTP VLAN enhancement allows you to preconfigure MSTP topologies before the VLAN IDs associated with each instance exist on a switch.*

When you use preconfigured VLAN ID-to-MSTI topologies, ensure that MSTP switches remain in the same region by mapping all VLAN IDs used in the region to the same MSTIs on each regional switch.

Configuring MSTP Instances with the VLAN Range Option

For the switches covered in this guide, if you use the **spanning-tree instance** command with the VLAN range option, even if the range includes VLANs that are not currently present on the switch, the entire range of VLANs is configured. For example, if VLANs 1, 5, and 7 are currently present and you enter this command:

```
ProCurve(config)# spanning-tree instance 1 vlan 1-10
```

then all the VLANs from 1 through 10 are included, even those VLANs that are not present.

On other ProCurve switches, only the VLANs that are present will be included, that is, only VLANs 1, 5, and 7 would be included. The switch will map these VLANs to MSTP Instance 1, which results in a Configuration Digest that is not the same as the Configuration Digest for the switches running this enhancement. (See Figure 4-15 and Figure 4-16)

Figure 4-15 shows an example of an MSTP instance configured with the VLAN range option. All the VLANs are included in the instance whether they exist or not. Figure 4-16 shows an example of an MSTP instance configured on another ProCurve switch. Only VLANs 1, 5, and 7 are included in the instance.

```
ProCurve(config)# show spanning-tree mst-config

MST Configuration Identifier Information
 MST Configuration Name: MSTP1
 MST Configuration Revision: 1
 MST Configuration Digest: 0x51B7EBA6BEED8702D2BA4497D4367517

 IST Mapped VLANs :

 Instance ID Mapped VLANs
 -------- ---------------
 1        1-10
```

Figure 4-15. An Example of Mapping VLANs with the Range Option where all VLANs are Included

The Configuration Digest value shown in Figure 4-16 is not the same as in Figure 4-15, indicating that these switches do not operate in the same instance.

The Common Spanning Tree (CST) will still have the correct root associations.

```
ProCurve(config)# show spanning-tree mst-config

MST Configuration Identifier Information
 MST Configuration Name: MSTP1
 MST Configuration Revision: 1
 MST Configuration Digest: 0x89D3ADV471668D6D832F6EC4AA9CF4AA

 IST Mapped VLANs :

 Instance ID Mapped VLANs
 -------- ---------------
 1        1, 5, 7
```

Figure 4-16. Example of Mapping VLANs on Other ProCurve Switches

Operating Notes for the VLAN Configuration Enhancement

- Configuring MSTP on the switch automatically configures the Internal Spanning Tree (IST) instance and places all statically and dynamically configured VLANs on the switch into the IST instance. The spanning-tree instance vlan command creates a new MST instance and moves the VLANs you specify from the IST to the MSTI.

  You must map a least one VLAN ID to an MSTI when you create it. You cannot map a VLAN ID to more than one instance. You can create up to 16 MSTIs in a region.

- The **no** form of the spanning-tree instance vlan command removes one or more VLANs from the specified MSTI. If no VLANs are specified, the **no** form of the command deletes the specified MSTI.

  When you remove a VLAN from and MSTI, the VLAN returns to the IST instance, where it can remain or be re-assigned to another MSTI configured in the region.

- If you enter the spanning-tree instance vlan command before a static or dynamic VLAN is configured on the switch to preconfigure VLAN ID-to-MSTI mappings, no error message is displayed. Later, each newly configured VLAN that has already been associated with an MSTI is automatically assigned to the MSTI.

  This new default behavior differs from automatically including configured (static and dynamic) VLANs in the IST instance and requiring you to manually assign individual static VLANs to an MSTI.

- The valid VLAN IDs that you can map to a specified MSTI are from 1 to 4094. The VLAN ID-to-MSTI mapping does not require a VLAN to be already configured on the switch. The MSTP VLAN enhancement allows you to preconfigure MSTP topologies before the VLAN IDs associated with each instance exist on a switch.

- When you use preconfigured VLAN ID-to-MSTI topologies, ensure that MSTP switches remain in the same region by mapping all VLAN IDs used in the region to the same MSTIs on each regional switch.

- When you upgrade switch software to release K.13.XX and later, the existing MSTP topology configuration is automatically saved. All existing VLAN ID-to-MSTI assignments are maintained on a switch for uninterrupted MSTP network operation.

How to Save Your Current Configuration

You can save your current configuration before updating to a new version of software by following these steps:

1. Enter the **show config files** command to display your current configuration files, as shown in Figure 4-17.

```
ProCurve(config)# show config files

Configuration files:

 id | act pri sec | name
 ---+-------------+--------------------
  1 |  *   *   *  | config1
  2 |             | config2
  3 |             |
```

Figure 4-17. An Example of the show config files Command Output

2. To save a configuration file for software version K.12.43, enter this command:

   ```
   ProCurve(config)# copy config config1 config
   configK1243.cfg
   ```

   You can choose any name for the saved configuration file that you prefer.

3. Display the configuration files as shown in Figure 4-18. You will see your newly created configuration file listed.

```
ProCurve(config)# show config files

Configuration files:

 id | act pri sec | name
 ---+-------------+----------------------
  1 |  *   *   *  | config1
  2 |             | config2
  3 |             | configK1243.cfg
```

Figure 4-18. A Config File for the Current Software Version is Created

4. Now update your switch to the desired version, for example, K.12.51. Enter the **show flash** command to see the results. The switch is now running the software version K.12.51.

```
ProCurve(config)# show flash

Image           Size(Bytes)   Date   Version  Build #
-----           ----------  -------- -------  -------
Primary Image   : 6771179   04/17/08 K.12.51    304
Secondary Image : 7408949   11/06/08 K.12.43    123
Boot Rom Version: K.12.12
Default Boot    : Primary
```

Figure 4-19. Show Flash Command after Upgrading the Switch to a New Version of the Software (K.12.51)

5. If you want to run the prior software version, K.12.43 in this example, enter this command:

   ```
   ProCurve(config)# boot system flash secondary config
   configK1243.cfg
   ```

   After rebooting, the switch is running software version K.12.43 and is using the configuration file that you saved for this software version, configK1243.cfg.

You can also save the K.12.43 configuration file on a TFTP server. If you wanted to reload the K.12.43 version of the software again, reload the configuration file before you do the reload.

Displaying MSTP Statistics and Configuration

Command	Page
MSTP Statistics:	
show spanning-tree [< *port-list* >]	below
show spanning-tree [< *port-list* >] detail	4-56
show spanning-tree instance < ist | 1..16 >	4-57
MSTP Configuration	
show spanning-tree [*port-list*] config	4-58
show spanning-tree [port-list] config instance < ist | 1..16 >	4-59
show spanning-tree mst-config	4-60
show spanning-tree pending < < instance | ist > | mst-config >	4-61

SNMP MIB Support for MSTP. MSTP is a superset of the STP/802.1D and RSTP/802.1w protocols and uses the MIB objects defined for these two protocols.

Displaying Global MSTP Status

The following commands display the MSTP statistics for the connections between MST regions in a network.

Syntax: show spanning-tree

This command displays the switch's global and regional spanning-tree status, plus the per-port spanning-tree operation at the regional level. Note that values for the following parameters appear only for ports connected to active devices: **Designated Bridge**, **Hello Time**, **PtP**, *and* **Edge**.

Syntax: show spanning-tree < *port-list* >

This command displays the spanning-tree status for the designated port(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, you would use this command: **show spanning-tree a20-a42,trk1**

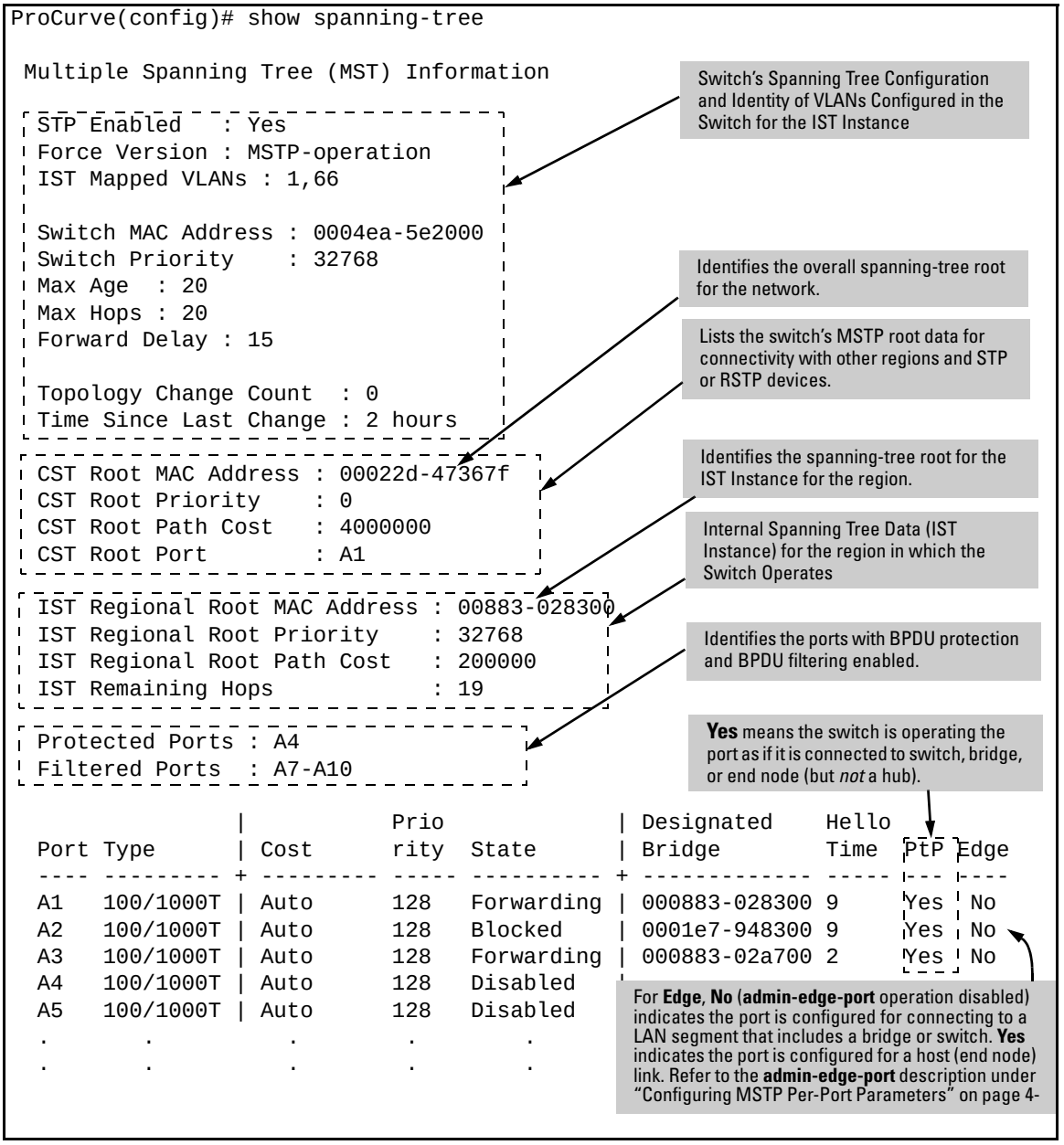

ProCurve(config)# show spanning-tree
Multiple Spanning Tree (MST) Information
STP Enabled : Yes
Force Version : MSTP-operation
IST Mapped VLANs : 1,66
Switch MAC Address : 0004ea-5e2000
Switch Priority : 32768
Max Age : 20
Max Hops : 20
Forward Delay : 15
Topology Change Count : 0
Time Since Last Change : 2 hours
CST Root MAC Address : 00022d-47367f
CST Root Priority : 0
CST Root Path Cost : 4000000
CST Root Port : A1
IST Regional Root MAC Address : 00883-028300
IST Regional Root Priority : 32768
IST Regional Root Path Cost : 200000
IST Remaining Hops : 19
Protected Ports : A4
Filtered Ports : A7-A10
Port Type | Cost Prio rity State | Designated Bridge Hello Time PtP Edge
A1 100/1000T | Auto 128 Forwarding | 000883-028300 9 Yes No
A2 100/1000T | Auto 128 Blocked | 0001e7-948300 9 Yes No
A3 100/1000T | Auto 128 Forwarding | 000883-02a700 2 Yes No
A4 100/1000T | Auto 128 Disabled
A5 100/1000T | Auto 128 Disabled
Switch's Spanning Tree Configuration and Identity of VLANs Configured in the Switch for the IST Instance
Identifies the overall spanning-tree root for the network.
Lists the switch's MSTP root data for connectivity with other regions and STP or RSTP devices.
Identifies the spanning-tree root for the IST Instance for the region.
Internal Spanning Tree Data (IST Instance) for the region in which the Switch Operates
Identifies the ports with BPDU protection and BPDU filtering enabled.
Yes means the switch is operating the port as if it is connected to switch, bridge, or end node (but not a hub).
For Edge, No (admin-edge-port operation disabled) indicates the port is configured for connecting to a LAN segment that includes a bridge or switch. Yes indicates the port is configured for a host (end node) link. Refer to the admin-edge-port description under "Configuring MSTP Per-Port Parameters" on page 4-

Figure 4-20. Example of Common Spanning Tree Status

Displaying Detailed Port Information

The following commands display the MSTP statistics for the connections between MST regions in a network.

Syntax: show spanning-tree detail

This command displays additional parameters concerning the common spanning tree (CST) ports.

Syntax: show spanning-tree < *port-list* > detail

This command displays detailed spanning-tree status for the designated port(s).

.

```
ProCurve# show spanning-tree a9 detail

 Status and Counters - CST Port(s) Detailed Information

 Port                        : A9
 Status                      : Up
 BPDU Filtering              : Yes
 Errant BPUDUs received      : 65
 MST Region Boundary         : Yes
 External Path Cost          : 200000
 External Root Path Cost     : 420021
 Administrative Hello Time   : Use Global
 Operational Hello Time      : 2
 AdminEdgePort               : No
 OperEdgePort                : No
 AdminPointToPointMAC        : Force-True
 OperPointToPointMAC         : Yes
 Aged BPDUs Count            : 0
 Loop-back BPDUs Count       : 0
 TC ACK Flag Transmitted     : 0
 TC ACK Flag Received        : 0

 MST         MST         CFG         CFG         TCN         TCN
 BPDUs Tx    BPDUs Rx    BPDUs Tx    BPDUs Rx    BPDUs Tx    BPDUs Rx
 ----------- ----------- ----------- ----------- ----------- ---------
 8           28          0           0           0           0
```

Gives information concerning the Common Spanning Tree (CST) only. Use the show spanning-tree instance commands to view counters pertaining to particular IST instances.

Figure 4-21. Example of CST Port Information using Show Spanning Tree Detail Command

Note This command gives information about the CST only. To view details of specific MST Instances, use the **show spanning tree instance** commands.

Displaying Status for a Specific MST Instance

The following commands display the MSTP statistics for a specified MST instance.

Syntax: show spanning-tree instance < ist | 1..16 >

This command displays the MSTP statistics for either the IST instance or a numbered MST instance running on the switch.

Syntax: show spanning-tree instance < ist | 1..16 > detail

This command displays status on all active ports for a specific instance of MSTP.

Syntax: show spanning-tree < *port-list* > instance < ist | 1..16 > detail

This command displays detailed status for the designated port(s) for a specific instance of MSTP.

```
Switch-1(config)# show spanning-tree instance 1

 MST Instance Information

  Instance ID : 1
  Mapped VLANs : 11,22

  Switch Priority          : 32768

  Topology Change Count    : 4
  Time Since Last Change   : 6 secs

  Regional Root MAC Address : 0001e7-948300
  Regional Root Priority    : 32768
  Regional Root Path Cost   : 400000
  Regional Root Port        : A1
  Remaining Hops            : 18
                                                                  Designated
  Port Type       Cost       Priority Role       State          Bridge
  ---- ---------- ---------- -------- ---------- ---------- --------------
  A1   10/100TX   200000     128      Root       Forwarding 000883-028300
  A2   10/100TX   200000     128      Designated Forwarding 000883-02a700
  A3   10/100TX   200000     112      Designated Forwarding 000883-02a700
  A4   10/100TX   Auto       128      Disabled   Disabled
   .       .        .         .          .          .
   .       .        .         .          .          .
   .       .        .         .          .          .
```

Figure 4-22. Example of MSTP Statistics for a Specific Instance on an MSTP Switch

Displaying the MSTP Configuration

Displaying the Global MSTP Configuration. This command displays the switch's basic and MST region spanning-tree configuration, including basic port connectivity settings.

Syntax: show spanning-tree config

The upper part of this output shows the switch's global spanning-tree configuration that applies to the MST region. The port listing shows the spanning-tree port parameter settings for the spanning-tree region operation (configured by the **spanning-tree < *port-list* >** *command). For information on these parameters, refer to "Configuring MSTP Per-Port Parameters" on page 4-26.*

Syntax: show spanning-tree < *port-list* > config

This command shows the same data as the above command, but lists the spanning-tree port parameter settings for only the specified port(s) and/or trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, use this command: **show spanning-tree a20-a24,trk1 config**

```
Switch-2(config)# show spanning-tree config

 Multiple Spanning Tree (MST) Configuration Information

  STP Enabled [No] : Yes
  Force Version [MSTP-operation] : MSTP-operation

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1          Switch Priority : 32768
  Forward Delay [15] : 15                 Hello Time [2] : 2
  Max Age [20] : 20                       Max Hops [20] : 20

  Port  Type       | Cost      Priority Edge Point-to-Point MCheck Hello Time
  ----- ---------- + --------- -------- ---- -------------- ------ ----------
  A3    10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
  A4    10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
   :        :            :        :        :              :      :
   :        :            :        :        :              :      :
   .        .            .        .        .              .      .
  A20   10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
  A21   10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
  A22   10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
  A23   10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
  A24   10/100TX   | Auto      128      Yes  Force-True     Yes    Use Global
  Trk1             | Auto      128      Yes  Force-True     Yes    Use Global
```

Global Priority

Global Hello Time

Per-Port Hello Time (Overrides Global Hello-Time on individual ports.)

Per-Port Priority

Figure 4-23. Example of Displaying the Switch's Global Spanning-Tree Configuration

Displaying Per-Instance MSTP Configurations. These commands displays the per-instance port configuration and current state, along with instance identifiers and regional root data.

Syntax: show spanning-tree config instance < ist | 1..16 >

The upper part of this output shows the instance data for the specified instance. The lower part of the output lists the spanning-tree port settings for the specified instance.

Syntax: show spanning-tree < *port-list* > config instance < ist | 1..16 >

This command shows the same data as the above command, but lists the spanning-tree port parameter settings for only the specified port(s) and/or trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, use this command:

show spanning-tree a20-a24,trk1 config instance 1

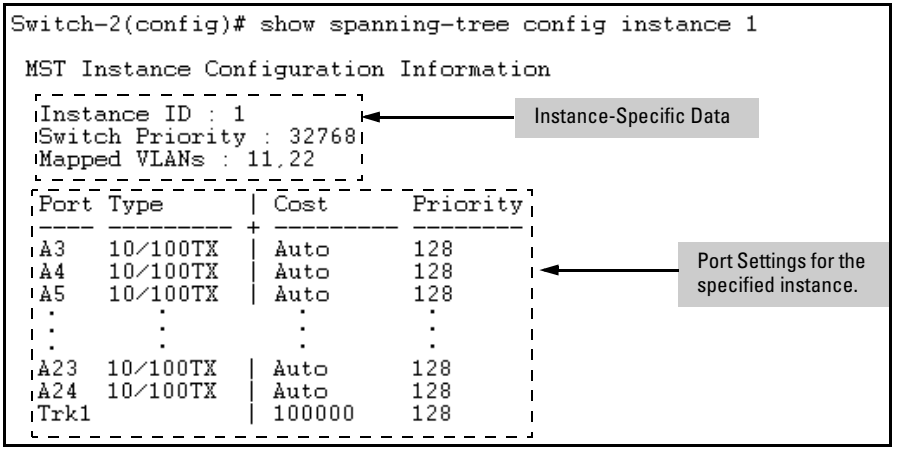

Switch-2(config)# show spanning-tree config instance 1
MST Instance Configuration Information
Instance ID : 1
Switch Priority : 32768
Mapped VLANs : 11,22
Instance-Specific Data
Port Type | Cost Priority
A3 10/100TX | Auto 128
A4 10/100TX | Auto 128
A5 10/100TX | Auto 128
A23 10/100TX | Auto 128
A24 10/100TX | Auto 128
Trk1 | 100000 128
Port Settings for the specified instance.

Figure 4-24. Example of the Configuration Listing for a Specific Instance

Displaying the Region-Level Configuration in Brief. This command output is useful for quickly verifying the allocation of VLANs in the switch's MSTP configuration and for viewing the configured region identifiers.

Syntax: show spanning-tree mst-config

This command displays the switch's regional configuration.

Note: *The switch computes the* ***MSTP Configuration Digest*** *from the VID to MSTI configuration mappings on the switch itself. As required by the 802.1s standard, all MSTP switches within the same region must have the same VID to MSTI assignments, and any given VID can be assigned to either the IST or one of the MSTIs within the region. Thus, the MSTP Configuration Digest must be identical for all MSTP switches intended to belong to the same region. When comparing two MSTP switches, if their Digest identifiers do not match, then they cannot be members of the same region.*

```
Switch-2(config)# show spanning-tree mst-config

 MST Configuration Identifier Information

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1
  MST Configuration Digest : 0xDAD6A13EC5141980B7EBDA71D8991E7C

  IST Mapped VLANs : 1,66

  Instance ID Mapped VLANs
  ----------- ------------------------------------------------
  1           11,22
  2           33,44,55
```

Refer to the "Note", above.

Figure 4-25. Example of a Region-Level Configuration Display

Displaying the Pending MSTP Configuration. This command displays the MSTP configuration the switch will implement if you execute the spanning-tree pending apply command (Refer to "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another" on page 4-44.)

Syntax: show spanning-tree pending < instance | mst-config >

instance < 1..16 | ist >

Lists region, instance I.D. and VLAN information for the specified, pending instance.

mst-config

Lists region, IST instance VLAN(s), numbered instances, and assigned VLAN information for the pending MSTP configuration.

```
ProCurve# show spanning-tree pending instance 1

 Pending MST Instance Configuration Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 10
  Instance ID : 1
  Mapped VLANs : 1,22



Switch-1(config)# show spanning-tree pending mst-config

 Pending MST Configuration Identifier Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 10

  IST Mapped VLANs : 11,33

  Instance ID Mapped VLANs
  ----------- ----------------------------------------------
  1           1,22
```

Figure 4-26. Example of Displaying a Pending Configuration

Troubleshooting an MSTP Configuration

Command	Page
show spanning-tree root-history	4-62
show spanning-tree debug counters	4-65
show spanning-tree debug-counters instance < *instance-id* >	4-66
show spanning-tree debug-counters instance < *instance-id* > ports <*port-list*>	4-68

This section describes the **show spanning-tree** commands that you can use to monitor, troubleshoot, and debug the operation of a multiple-instance spanning-tree configuration in your network.

Note that the **show spanning-tree** commands described in this section allow you to troubleshoot MSTP activity in your network by focusing on increasingly specific levels of operation. For example, you can display debug information for:

- All MST instances
- All ports used in one MST instance
- A specific port or several ports used in one MST instance

Also, you can display the change history for the root (bridge) switch used as the single forwarding path for:

- All MST regions, STP bridges, and RSTP bridges in an STP network
- All VLANs on MSTP switches in a region
- All VLANs on MSTP switches in an MST instance

Displaying the Change History of Root Bridges

The **show spanning-tree root-history** command allows you to display change history information (up to 10 history entries) for a specified root bridge in any of the following MSTP topologies:

- Common Spanning Tree (**cst**): Provides connectivity in a bridged network between MST regions, STP LANs, and RSTP LANs.
- Internal Spanning Tree (**ist**): Provides connectivity within an MST region for VLANs associated with the default Common and Internal Spanning Tree (CIST) instance in your network (VLANs that have not been mapped to an MST instance).

- MST Instance (**mst**): Connects all static and (starting from release 13.*x*.*x*) dynamic VLANs assigned to a multiple spanning-tree instance.

Syntax: show spanning-tree root-history <cst | ist | mst <*instance-id*>>

This command displays the change history for the root bridge in the specified MSTP topology.

The **cst** *parameter displays the change history for the root bridge of a spanning-tree network, including MST regions and STP and RSTP bridges.*

The **ist** *parameter displays the change history for the root bridge in the IST instance of an MST region.*

The **mst** *<instance-id> parameter displays the change history for the root bridge in an MST instance, where <instance-id> is an ID number from 1 to 16.*

Use the **show spanning-tree root-history** command to view the number and dates of changes in the assignment of a root bridge. Possible intrusion into your MST network may occur if an unauthorized external device gains access to a spanning tree by posing as the root device in a topology. To prevent an MST port connected to the device from being selected as the root port in a topology, use the **spanning-tree root-guard** command.

The following examples show sample output of the **show spanning-tree root-history** command for different MSTP topologies. Note that in each example, the root bridge ID is displayed in the format:
<*priority*:*mac-address*>

Where:

- <*priority*> is the MSTP switch priority calculated for one of the following:
  - The IST (regional) root switch using the **spanning-tree priority** command
  - An MSTI root switch using the **spanning-tree instance priority** command
- <*mac-address*> is the MAC address of the root (bridge) switch.

```
ProCurve(config)# show spanning-tree root-history cst

 Status and Counters - CST Root Changes History

  MST Instance ID        : 0
  Root Changes Counter   : 2
  Current Root Bridge ID : 32768:000883-024500

  Root Bridge ID      Date     Time
  ------------------- -------- --------
  32768:000883-024500 02/09/07 17:40:59
  36864:001279-886300 02/09/07 17:40:22
```

Identifies the root bridge of the common spanning tree in a bridged network that connects different MST regions and STP or RSTP devices.

Figure 4-27. Example of show spanning-tree root-history cst Command Output

```
ProCurve(config)# show spanning-tree root-history ist

 Status and Counters - IST Regional Root Changes History

  MST Instance ID        : 0
  Root Changes Counter   : 2
  Current Root Bridge ID : 32768:000883-024500

  Root Bridge ID      Date     Time
  ------------------- -------- --------
  32768:000883-024500 02/09/07 17:40:59
  36864:001279-886300 02/09/07 17:40:22
```

Identifies the root bridge of the internal spanning tree in an MST region.

Figure 4-28. Example of show spanning-tree root-history ist Command Output

```
ProCurve(config)# show spanning-tree root-history mst 2

 Status and Counters - MST Instance Regional Root Changes History

  MST Instance ID        : 2
  Root Changes Counter   : 2
  Current Root Bridge ID : 32770:000883-024500

  Root Bridge ID      Date     Time
  ------------------- -------- --------
  32770:000883-024500 02/09/07 17:40:59
  32770:001279-886300 02/09/07 17:40:22
```

Identifies the root bridge of an MST instance in an MST region.

Figure 4-29. Example of show spanning-tree root-history msti Command Output

Displaying Debug Counters for All MST Instances

The **show spanning-tree debug-counters** command allows you to display the aggregate values of all MSTP debug counters that are maintained on a switch. These aggregate values are a summary of the information collected from all ports and from all spanning-tree instances that forward traffic on switch ports.

Use the displayed diagnostic information to globally monitor MSTP operation on a per-switch basis.

Syntax: show spanning-tree debug-counters

This command displays debug counters for MSTP activity on all ports configured for VLANs used in spanning-tree instances.

The following example shows sample output of the **show spanning-tree debug-counters** command for all ports. For a description of each counter, refer to Table 4-1 on page 4-70.

```
ProCurve(config)# show spanning-tree debug-counters

 Status and Counters - MSTP Bridge Common Debug Counters Information

  Counter Name                      Aggregated Value Collected From
  --------------------------------- ---------------- --------------
  Invalid BPDUs                     0                CIST
  Errant BPDUs                      170927           CIST
  MST Config Error BPDUs            0                CIST
  Looped-back BPDUs                 0                CIST
  Starved BPDUs/MSTI MSGs           0                CIST/MSTIs
  Exceeded Max Age BPDUs            0                CIST
  Exceeded Max Hops BPDUs/MSTI MSGs 0                CIST/MSTIs
  Topology Changes Detected         2                CIST/MSTIs
  Topology Changes Tx               6                CIST/MSTIs
  Topology Changes Rx               4                CIST/MSTIs
  Topology Change ACKs Tx           0                CIST
  Topology Change ACKs Rx           0                CIST
  TCN BPDUs Tx                      0                CIST
  TCN BPDUs Rx                      0                CIST
  CFG BPDUs Tx                      0                CIST
  CFG BPDUs Rx                      0                CIST
  RST BPDUs Tx                      0                CIST
  RST BPDUs Rx                      0                CIST
  MST BPDUs/MSTI MSGs Tx            10               CIST/MSTIs
  MST BPDUs/MSTI MSGs Rx            341802           CIST/MSTIs
```

Figure 4-30. Example of show spanning-tree debug-counters Command Output

Displaying Debug Counters for One MST Instance

The **show spanning-tree debug-counters instance** command allows you to display the aggregate values of all MSTP debug counters maintained on a switch for a specified spanning-tree instance. These aggregate values are a summary of information collected from all ports that have VLANs assigned to the specified instance.

Use the **show spanning-tree debug-counters instance** command to troubleshoot the global MSTP diagnostic information displayed in **show spanning-tree debug-counters** command output when you suspect unauthorized MSTP activity in a specific MST instance.

Syntax: show spanning-tree debug-counters instance <*instance-id*>

This command displays debug counters for MSTP activity on all ports configured for VLANs in the specified MST instance. The valid values for **instance** <*instance-id*> *are from 0 to 16:*

- *0 specifies the default MST (CIST) instance.*
- *1 to 16 specify a multiple spanning-tree (MST) instance.*

The following example shows sample output of the **show spanning-tree debug-counters instance** command when applied to the Common and Internal Spanning Tree (CIST) instance (default MST instance 0) in the network. For a description of each counter, refer to Table 4-1 on page 4-70.

```
ProCurve(config)# show spanning-tree debug-counters instance 0

 Status and Counters - CIST Common Debug Counters Information

  MST Instance ID : 0

  Counter Name                     Aggregated Value Collected From
  -------------------------------- ---------------- --------------
  Invalid BPDUs                    0                Ports
  Errant BPDUs                     172603           Ports
  MST Config Error BPDUs           0                Ports
  Looped-back BPDUs                0                Ports
  Starved BPDUs                    0                Ports
  Exceeded Max Age BPDUs           0                Ports
  Exceeded Max Hops BPDUs          0                Ports
  Topology Changes Detected        1                Ports
  Topology Changes Tx              3                Ports
  Topology Changes Rx              2                Ports
  Topology Change ACKs Tx          0                Ports
  Topology Change ACKs Rx          0                Ports
  TCN BPDUs Tx                     0                Ports
  TCN BPDUs Rx                     0                Ports
  CFG BPDUs Tx                     0                Ports
  CFG BPDUs Rx                     0                Ports
  RST BPDUs Tx                     0                Ports
  RST BPDUs Rx                     0                Ports
  MST BPDUs Tx                     5                Ports
  MST BPDUs Rx                     172577           Ports
```

Figure 4-31. Example of show spanning-tree debug-counters instance Command Output for All Ports in the CIST Instance

Displaying Debug Counters for Ports in an MST Instance

The **show spanning-tree debug-counters instance ports** command allows you to display the aggregate values of all MSTP debug counters maintained on one or more ports used by a specified spanning-tree instance. These aggregate values are a summary of information collected from the specified ports that have VLANs assigned to the specified instance.

Use the **show spanning-tree debug-counters instance ports** command to troubleshoot at a finer level the more general MSTP diagnostic information displayed in **show spanning-tree debug-counters instance** command output when you suspect unauthorized MSTP activity on one or more MST ports in an MST instance.

Syntax: show spanning-tree debug-counters instance <*instance-id*> ports <*port-list*>

This command displays debug counters for MSTP activity on the specified ports configured for VLANs in the specified MST instance.

The valid values for **instance** <*instance-id*> *are from 0 to 16:*

- *0 specifies the default MST (CIST) instance.*
- *1 to 16 specify an MST instance.*

The **ports** <*port-list*> *parameter specifies one or more MST ports or trunk ports.*

In the port list, enter a series of ports by separating the first and last ports in the series with a dash (-); for example, **a2-a8** *or* **trk1-trk3**. *Separate individual ports and series of ports with a comma; for example,* **a2-a8**, **a20**, **trk1, trk4-trk5**.

The following examples shows sample output of the **show spanning-tree debug-counters instance ports** command for both the CIST (default MST instance 0) and an MST instance (instance 2) on port A15. For a description of each counter, refer to Table 4-1 on page 4-70.

```
ProCurve(config)# show spanning-tree debug-counters instance 0 ports a15

 Status and Counters - CIST Port(s) Debug Counters Information

  MST Instance ID : 0
  Port : A15

  Counter Name                Value      Last Updated
  --------------------------- ---------- ----------------
  Invalid BPDUs               0
  Errant BPDUs                0
  MST Config Error BPDUs      0
  Looped-back BPDUs           0
  Starved BPDUs               0
  Exceeded Max Age BPDUs      0
  Exceeded Max Hops BPDUs     0
  Topology Changes Detected   1          02/09/07 17:40:59
  Topology Changes Tx         3          02/09/07 17:41:03
  Topology Changes Rx         2          02/09/07 17:41:01
  Topology Change ACKs Tx     0
  Topology Change ACKs Rx     0
  TCN BPDUs Tx                0
  TCN BPDUs Rx                0
  CFG BPDUs Tx                0
  CFG BPDUs Rx                0
  RST BPDUs Tx                0
  RST BPDUs Rx                0
  MST BPDUs Tx                5          02/09/07 17:41:03
  MST BPDUs Rx                173540     02/13/07 18:05:34
```

Figure 4-32. Example of show spanning-tree debug-counters instance ports Command Output for One Port in the CIST Instance

```
ProCurve(config)# show spanning-tree debug-counters instance 2 ports a15

 Status and Counters - MSTI Port(s) Debug Counters Information

  MST Instance ID : 2
  Port : A15

  Counter Name                Value      Last Updated
  --------------------------- ---------- ----------------
  Starved MSTI MSGs           0
  Exceeded Max Hops MSTI MSGs 0
  Topology Changes Detected   1          02/09/07 17:40:59
  Topology Changes Tx         3          02/09/07 17:41:03
  Topology Changes Rx         2          02/09/07 17:41:01
  MSTI MSGs Tx                5          02/09/07 17:41:03
  MSTI MSGs Rx                173489     02/13/07 18:03:52
```

Figure 4-33. Example of show spanning-tree debug-counters instance ports Command Output for One Port in an MST Instance

Field Descriptions in MSTP Debug Command Output

Table 4-1 contains descriptions of the debugging information displayed in the output of **show spanning-tree debug-counters** commands.

Table 4-1. MSTP Debug Command Output: Field Descriptions

Field	Description
Invalid BPDUs	Number of received BPDUs that failed standard MSTP (802.1Q-REV/D5.0 14.4) validation checks and were dropped. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Errant BPDUs	Number of received BPDUs that were dropped on a port that is configured to not expect BPDU packets. This counter is maintained by the CIST (default MST instance 0 in the network) on a per-port basis and is incremented each time a BPDU packet is received on a port configured with the BPDU filter to ignore incoming BPDU packets (**spanning-tree bpdu-filter** command) or the BPDU protection feature to disable the port when BPDU packets are received (**spanning-tree bpdu-protection** command).

Field	Description
MST Config Error BPDUs	Number of BPDUs received from a neighbor bridge with inconsistent MST configuration information. For example, BPDUs from a transmitting bridge may contain the same MST configuration identifiers (region name and revision number) and format selector as the receiving bridge, but the value of the Configuration Digest field (VLAN ID assignments to regional IST and MST instances) is different. This difference indicates a probable configuration error in MST region settings on the communicating bridges. The received BPDU is still processed by MSTP. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Looped-back BPDUs	Number of times a port has received self-sent BPDU packets as the result of an external loop condition in which the BPDUs were looped back to the originating transmission port. The received BPDU is still processed by MSTP and the port changes to a blocked state. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Starved BPDUs	Number of times that no BPDUs are received within the scheduled interval (three times the Hello Time value configured with the **spanning-tree hello-time** command) from a downstream CIST-designated peer port on the CIST root, alternate, or backup port. As a result, the "starved" port triggers a spanning-tree topology regeneration. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Starved MSTI MSGs	Number of times that no BPDUs are received within the scheduled interval (three times the Hello Time value configured with the **spanning-tree hello-time** command) from a downstream MSTI-designated peer port on the MSTI root, alternate, or backup port. As a result, the "starved" port triggers a spanning-tree topology regeneration. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Exceeded Max Age BPDUs	Number of times that a BPDU packet is received from a bridge external to the MST region with a Message Age value greater than the configured value of the Max Age parameter (**spanning-tree maximum age** command). This may occur if the receiving bridge is located too far from the root bridge (beyond the configured size of the spanning-tree domain on the root bridge) or if a BPDU packet with invalid root information is continuously circulating between bridges in a spanning-tree domain and needs to be aged out. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Exceeded Max Hops BPDUs	Number of times that a BPDU packet is received from a bridge internal to the MST region with a CIST Remaining Hops value less than or equal to 1. This may occur if the receiving bridge is located too far from the CIST regional root bridge (beyond the configured size of the MST region on the CIST regional root bridge) or if a PDU packet with invalid CIST regional root bridge information is continuously circulating between bridges in the MST Region and needs to be aged out. This counter is maintained by the CIST (default MST instance 0 in the region) on a per-port basis.

Field	Description
Exceeded Max Hops MSTI MSGs	Number of times that an MSTI MSG packet is received from a bridge internal to the MST region with an MSTI Remaining Hops value less than or equal to 1. This may occur if the receiving bridge is located too far from the MSTI regional root bridge (beyond the configured size of the MST region on the MSTI regional root bridge) or if a BPDU packet with invalid MSTI regional root bridge information is continuously circulating between bridges in an MST region and needs to be aged out. This counter is maintained on a per-MSTI per-port basis.
Topology Changes Detected	Number of times that a Topology Change event is detected by the CIST or MSTI port and the port triggers a topology change propagation throughout the network. A Topology Change event occurs when a non-edge port enters forwarding state. This counter is maintained on a per-CIST per-port and on a per-MSTI per-port basis.
Topology Changes Tx	Number of times that Topology Change information is propagated (sent out) through the port to the rest of the network. For a CIST port, the counter is the number of times that a CFG, RST or MST BPDU with the TC flag set is transmitted out of the port. For an MSTI port, the counter is the number of times that a MSTI configuration message with the TC flag set is transmitted out of the port. This counter is maintained on a per-CIST per-port and on a per-MSTI per-port bases.
Topology Changes Rx	Number of times that Topology Change information is received from the peer port. For a CIST port, the counter is the number of times that a CFG, RST or MST BPDU with the TC flag set is received. For an MSTI port, the counter is the number of times that an MSTI configuration message with the TC flag set is received. This counter is maintained on a per-CIST per-port and on a per-MSTI per-port basis.
Topology Change ACKs Tx	Number of times that the Topology Change acknowledgement is transmitted through the port (number of CFG, RST or MST BPDUs transmitted with the Topology Change Acknowledge flag set). This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
Topology Change ACKs Rx	Number of times the Topology Change acknowledgement is received on the port (number of CFG, RST or MST BPDUs received with the Topology Change Acknowledge flag set). This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
TCN BPDUs Tx	Number of Topology Change Notification BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
TCN BPDUs Rx	Number of Topology Change Notification BPDUs that are received on the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
CFG BPDUs Tx	Number of (802.1D) Configuration BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
CFG BPDUs Rx	Number of (802.1D) Configuration BPDUs that are received on the port. This counter maintained by the CIST (default MST instance 0) on a per-port basis.

Field	Description
RST BPDUs Tx	Number of (802.1w) RST BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
RST BPDUs Rx	Number of (802.1w) RST BPDUs that are received on the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
MST BPDUs Tx	Number of (802.1s) MST BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
MST BPDUs Rx	Number of (802.1s) MST BPDUs that are received on the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
MSTI MSGs Tx	Number of times that a configuration message for a specific MSTI was encoded in (802.1s) MST BPDUs that are transmitted through the port. This counter is maintained on a per-MSTI per-port basis.
MSTI MSGs Rx	Number of times that the MSTI detected a configuration message destined to the MSTI in (802.1s) MST BPDUs received on the port. This counter is maintained on a per-MSTI per-port basis.

Troubleshooting MSTP Operation

Table 4-2. Troubleshooting MSTP Operation

Problem	Possible Cause
Duplicate packets on a VLAN, or packets not arriving on a LAN at all.	The allocation of VLANs to MSTIs may not be identical among all switches in a region.
A switch intended to operate in a region does not receive traffic from other switches in the region.	An MSTP switch intended for a particular region may not have the same configuration name or region revision number as the other switches intended for the same region. The MSTP configuration name (**spanning-tree config-name** command) and MSTP configuration revision number (**spanning-tree config-revision** command) must be identical on all MSTP switches intended for the same region. Another possible cause is that the set of VLANs and VLAN ID-to-MSTI mappings (**spanning-tree instance vlan** command) configured on the switch may not match the set of VLANs and VLAN ID-to-MSTI mappings configured on other switches in the intended region.

Loop Protection

In cases where spanning tree cannot be used to prevent loops at the edge of the network, loop protection may provide a suitable alternative. Unlike spanning tree, however, loop protection is not a comprehensive loop detection feature and should only be enabled on untagged edge ports, that is, ports that connect to unmanaged switches and/or clients at the edge of the network.

The cases where loop protection might be chosen ahead of spanning tree to detect and prevent loops are as follows:

- **On ports with client authentication**. When spanning tree is enabled on a switch that use 802.1X, Web authentication, and MAC authentication, loops may go undetected. For example, spanning tree packets that are looped back to an edge port will not be processed because they have a different broadcast/multicast MAC address from the client-authenticated MAC address. To ensure that client-authenticated edge ports get blocked when loops occur, you should enable loop protection on those ports.
- **On ports connected to unmanaged devices**. Spanning tree cannot detect the formation of loops where there is an unmanaged device on the network that does not process spanning tree packets and simply drops them. Loop protection has no such limitation, and can be used to prevent loops on unmanaged switches.

Figure 4-34 shows examples where loop protection can be used.

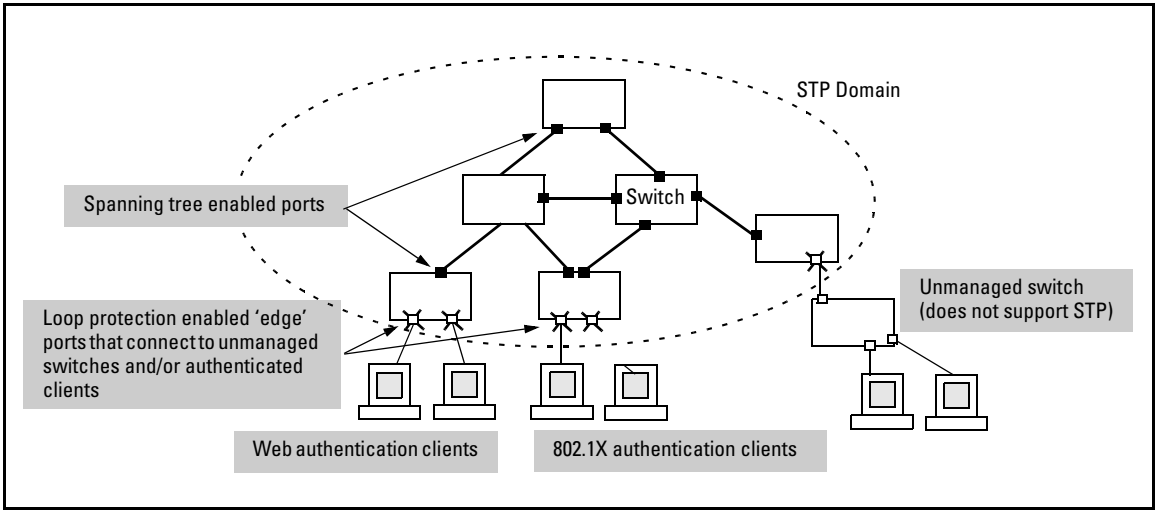

STP Domain
Spanning tree enabled ports
Switch
Unmanaged switch
(does not support STP)
Loop protection enabled 'edge' ports that connect to unmanaged switches and/or authenticated clients
Web authentication clients
802.1X authentication clients

Figure 4-34. Examples of Loop Protection Enabled in Preference to STP

Configuring Loop Protection

Loop protection provides protection against loops by transmitting loop protocol packets out of ports on which loop protection has been enabled. When the switch sends out a loop protocol packet and then receives the same packet on a port that has a **receiver-action** of **send-disable** configured, it shuts down the port from which the packet was sent.

To enable loop protection:

1. Enter the **loop-protect** command and specify the port(s) on which loop protection should be enabled. For example:

   ```
   ProCurve(config)# loop-protect 1-4
   ```

2. Specify a **receiver-action** of **send-disable** to shut down the port in the event of a loop. For example:

   ```
   ProCurve(config)# loop-protect 1 receiver-action send-
   disable
   ```

Syntax: [no] loop-protect <port-list> [receiver-action <send-disable | no-disable>] | [transmit-interval <1-10>] | [disable-timer <0-604800>] | [trap loop-detected]

Allows you to configure per-port loop protection on the switch.

[receiver-action <send-disable | no-disable>]

Sets the action to be taken when a loop is detected on the specified port or ports. The port that receives the loop protection packet determines what action is taken. If send-disable is configured, the port that transmitted the packet is disabled. If no-disable is configured, the port is not disabled when a loop is detected.
Default: send-disable

[trap loop-detected]

Allows you to configure loop protection traps for SNMP indicating when a loop has been detected on a port.

[disable-timer <0-604800>]

How long (in seconds) a port is disabled when a loop has been detected. A value of zero disables the auto re-enable function.
Default: Timer is disabled

[transmit-interval <1-10>]

Allows you to configure the time in seconds between the transmission of loop protection packets.
Default: 5 seconds

Notes

- The **receiver-action** option can be configured on a per-port basis and can only be enabled after loop protection has been enabled on the port. All other configuration options (**disable-timer**, **trap loop-detected**, and **transmit interval**) are global.
- The **trap** option refers to a SNMP trap.
- Regardless of how the **receiver-action** and **trap** options are configured, all detected loops will be logged in the switch's event log.
- The **no loop-protect** <*port*> command will not remove a receive-action configuration line from the running configuration unless this option is set to **receive-action send-disable**.

Viewing Loop Protection Status

To display information about ports with loop protection enabled, enter the **show loop-protect** command.

Syntax: show loop-protect <*port-list*>

Displays the loop protection status. If no ports are specified, the information is displayed only for the ports that have loop protection enabled.

For example:

```
ProCurve(config)# show loop-protect 1-4

 Status and Counters - Loop Protection Information

 Transmit Interval (sec)  : 5
 Port Disable Timer (sec) : 5
 Loop Detected Trap       : Enabled


      Loop        Loop      Loop       Time              Rx           Port
 Port Protection  Detected  Count      Since Last Loop   Action       Status
 ---- ----------- --------- ---------- ----------------- ------------ --------
 1    Yes          No        0                           send-disable Up
 2    Yes          No        0                           send-disable Up
 3    Yes          No        0                           send-disable Up
 4    Yes          No        0                           send-disable Up
```

Figure 4-35. Example of Show Loop-Protect Display

5

Switch Meshing

Contents

Introduction 5-2

Switch Meshing Fundamentals 5-4

Terminology 5-4

Operating Rules 5-5

Using a Heterogeneous Switch Mesh 5-7

Bringing Up a Switch Mesh Domain 5-8

Further Operating Information 5-8

Configuring Switch Meshing 5-9

Preparation 5-9

Menu: To Configure Switch Meshing 5-9

CLI: To View and Configure Switch Meshing 5-12

Viewing Switch Mesh Status 5-12

CLI: Configuring Switch Meshing 5-14

Operating Notes for Switch Meshing 5-15

Flooded Traffic 5-16

Unicast Packets with Unknown Destinations 5-17

Spanning Tree Operation with Switch Meshing 5-17

Filtering/Security in Meshed Switches 5-20

IP Multicast (IGMP) in Meshed Switches 5-20

Static VLANs 5-20

Dynamic VLANs 5-21

Jumbo Packets 5-21

Mesh Design Optimization 5-22

Other Requirements and Restrictions 5-23

Introduction

Switch meshing is a load-balancing technology that enhances reliability and performance in these ways:

- Provides significantly better bandwidth utilization than either Spanning Tree Protocol (MSTP) or standard port trunking.
- Uses redundant links that remain open to carry traffic, removing any single point of failure for disabling the network, and allowing quick responses to individual link failures. This also helps to maximize investments in ports and cabling.
- Unlike trunked ports, the ports in a switch mesh can be of different types and speeds (10 and 100 Mbps, gigabit, and 10 gigabit). For example, a 10Base-FL port and a 1GB port can be included in the same switch mesh.

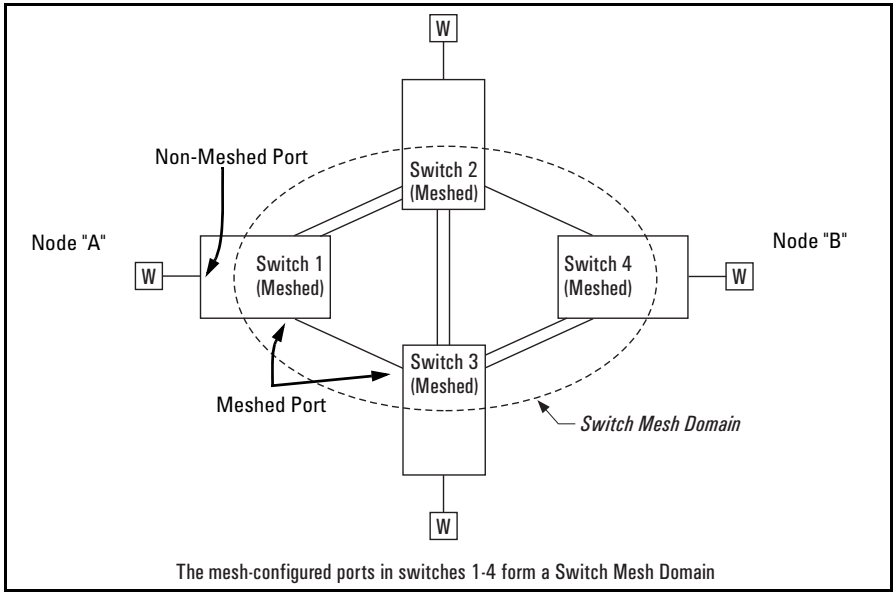


Figure 5-1. Example of Switch Meshing

Finding the Fastest Path. Using multiple switches redundantly linked together to form a *meshed switch domain*, switch meshing dynamically distributes traffic across load-balanced switch paths by seeking the fastest paths for new traffic between nodes. In actual operation, the switch mesh periodically determines the best (lowest latency) paths, then assigns these paths as the need arises. The path assignment remains until the related MAC address entry times out. The mesh sees later traffic between the same nodes as new traffic, and may assign a different path, depending on conditions at the time. For example, at one time the best path from node A to node B is through switch 2. However, if traffic between node A and node B ceases long enough for the path assignment to age out, then the next time node A has traffic for node B, the assigned path between these nodes may be through switch 3 if network conditions have changed significantly.

Note

The **mac-age-time** parameter determines how long an inactive path assignment remains in memory. Refer to "System Information" in the chapter titled "Interface Access and System Information" in the *Management and Configuration Guide* for your switch.

Because Redundant Paths Are Active, Meshing Adjusts Quickly to Link Failures. If a link in the mesh fails, the fast convergence time designed into meshing typically has an alternate route selected in less than a second for traffic that was destined for the failed link.

Meshing Allows Scalable Responses to Increasing Bandwidth Demand. As more bandwidth is needed in a LAN backbone, another switch and another set of links can be added. This means that bandwidth is not limited by the number of trunk ports allowed in a single switch.

Meshing Features

Feature	Default	Menu	CLI	Web
Viewing a mesh configuration	n/a	5-9	5-12	n/a
Configuring a Switch Mesh	n/a	5-9	5-14	n/a

Switch Meshing Fundamentals

Terminology

Switch Mesh Domain. This is a group of meshed switch ports exchanging meshing protocol packets. Paths between these ports can have multiple redundant links without creating broadcast storms.

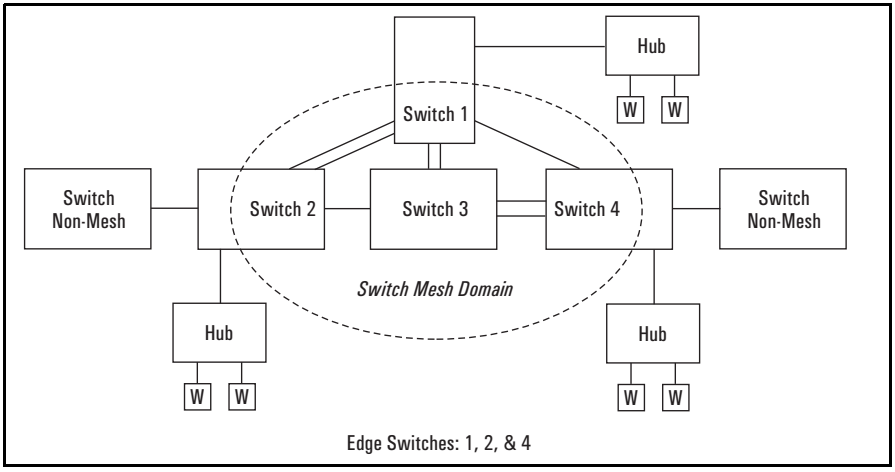


Figure 5-2. Example of a Switch Mesh Domain in a Network

Edge Switch. This is a switch that has some ports in the switch meshing domain and some ports outside of the domain. (See figure 5-2, above.)

Operating Rules

(See also "Mesh Design Optimization" on page 5-22.)

- A meshed switch can have some ports in the meshed domain and other ports outside the meshed domain. That is, ports within the meshed domain must be configured for meshing, while ports outside the meshed domain must not be configured for meshing.
- Meshed links must be point-to-point switch links.
- On any switch, all meshed ports belong to the same mesh domain.
- A switch can have up to 24 meshed ports.
- A mesh domain can include up to 12 switches.
- Up to five inter-switch, meshed hops are allowed in the path connecting two nodes through a switch mesh domain. A path of six or more meshed hops between two nodes is unusable. However, in most mesh topologies, there would normally be a shorter path available, and paths of five hops or fewer through the same mesh will continue to operate.
- Hub links between meshed switch links are not allowed.
- If the switch has multiple static VLANs and you configure a port for meshing, the port becomes a tagged member of all such VLANs . If you remove a port from meshing, it becomes an untagged member of only the default VLAN.
- A port configured as a member of a *static* trunk (LACP or Trunk) cannot also be configured for meshing.
- If a port belongs to a *dynamic* LACP trunk and you impose meshing on the port, it automatically ceases to be a member of the dynamic trunk.
- Meshing is not supported on ports configured with 802.1X access control.
- On a port configured for meshing, if you subsequently remove meshing from the port's configuration and reboot the switch, the port returns to its default configuration. (It *does not* revert to any non-default configuration it had before being configured for meshing).
- In a given mesh domain, switches in the same product family must run the same switch software version. For example, if you update the software version on one 8212zl switch, then you must update the software version on any other 8212zl switch in the mesh. ProCurve recommends that you always use the most recent software version available for the switches in your network.
- Meshing and routing features (such as IP routing, RIP, and OSPF) are not supported at the same time on a switch. To enable meshing, you must disable routing, and vice versa. *Switch meshing and routing cannot be enabled at the same time.*

- The spanning-tree configuration must be the same for all switches in the mesh (enabled or disabled). If spanning tree is enabled in the mesh, it must be the same version on all switches in the mesh: 802.1D, 802.1w, or 802.1s.
- If a switch in the mesh has GVRP enabled, then all switches in the mesh must have GVRP enabled. Otherwise, traffic on a dynamic VLAN may not pass through the mesh.
- If a switch in the mesh has a particular static vlan configured, then all switches in the mesh must have that static vlan configured.
- If a switch in the mesh has IGMP enabled, then all switches in the mesh must have IGMP enabled.
- If a switch in the mesh has LLDP enabled, then all switches in the mesh must have LLDP enabled.
- After adding or removing a port from the mesh, you must save the current configuration and reboot the switch in order for the change to take effect.
- Multiple meshed domains require separation by either a non-meshed switch or a non-meshed link. For example:

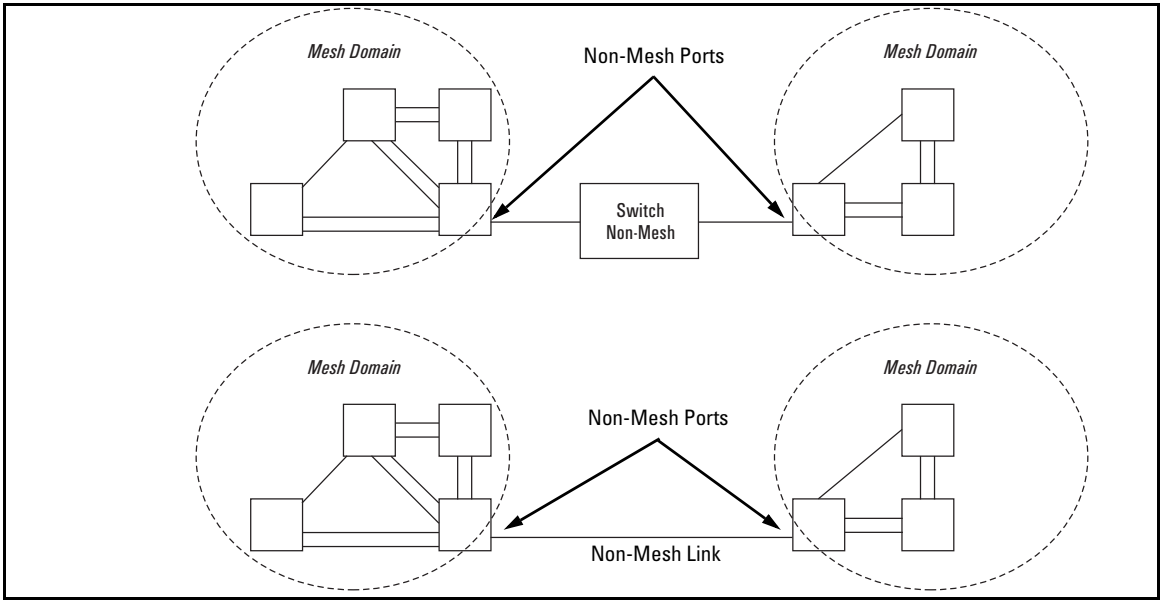


Figure 5-3. Example of Multiple Meshed Domains Separated by a Non-Mesh Switch or a Non-Mesh Link

- If GVRP is enabled, meshed ports in a switch become members of any dynamic VLANs created in the switch in the same way that they would if meshing was not configured in the switch. (For more on GVRP, refer to chapter 3, “GVRP”.)

Note

- A switch mesh domain (figure 5-1 on page 5-2) cannot include either a switch that is not configured for meshing, or a hub.
- Where a given pair of switches are linked with meshed ports, you must not also link the pair together through non-meshed ports unless you have also enabled STP, RSTP, or MSTP to prevent a loop from forming.

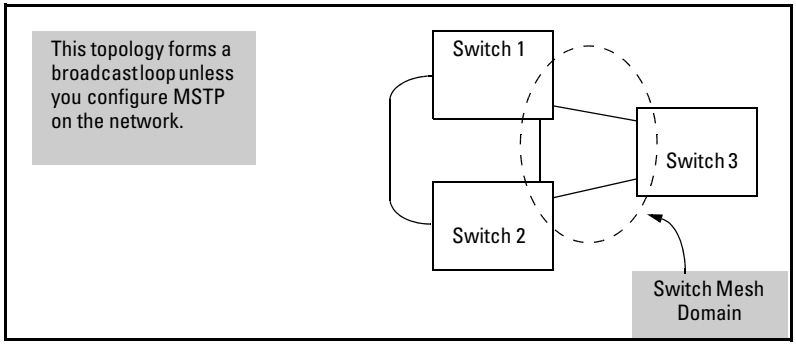

This topology forms a broadcast loop unless you configure MSTP on the network.
Switch 1
Switch 3
Switch 2
Switch Mesh Domain

Figure 5-4. Example of an Unsupported Topology

- The switch blocks traffic on a meshed port connected to a non-meshed port on another switch.
- Switch meshing does not allow trunked links (LACP or Trunk) between meshed ports.

Linking a non-mesh device or port into the mesh causes the meshed switch port(s) connected to that device to shut down.

Using a Heterogeneous Switch Mesh

You can use the switches covered in this guide with the ProCurve Series 5300xl switches in normal mode.

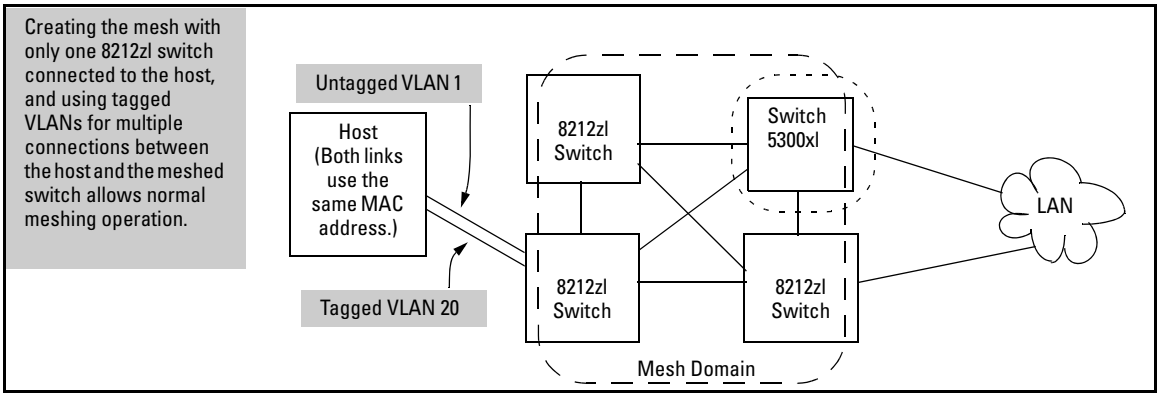

Creating the mesh with only one 8212zl switch connected to the host, and using tagged VLANs for multiple connections between the host and the meshed switch allows normal meshing operation.
Untagged VLAN 1
Host (Both links use the same MAC address.)
Tagged VLAN 20
8212zl Switch
Switch 5300xl
8212zl Switch
8212zl Switch
LAN
Mesh Domain

Figure 5-5. Example of a Supported Heterogeneous Topology in Normal Mode

Bringing Up a Switch Mesh Domain

When a meshed port detects a non-meshed port on the opposite end of a point-to-point connection, the link will be blocked. Thus, as you bring up switch meshing on various switches, you may temporarily experience blocked ports where meshed links should be running. These conditions should clear themselves after all switches in the mesh have been configured for meshing and their switches rebooted. To reduce the effect of blocked ports during bring-up, configure meshing and reboot the switches before installing the meshed switches in the network. Also, since adding (or removing) a meshed port requires a switch reboot to implement, you can avoid repeated system disruptions by waiting to implement the mesh until you have finished configuring meshing on all ports in your intended mesh domain.

Further Operating Information

Refer to “Operating Notes for Switch Meshing” on page 5-15.

Configuring Switch Meshing

Preparation

Before configuring switch meshing:

- Review the Operating Rules (page 5-5), and particularly the restrictions and requirements for using switch meshing in environments that include static trunks, multiple static VLANs, GVRP, IGMP, and MSTP.
- To avoid unnecessary system disruption, plan the mesh bring-up to minimize temporary port-blocking. (Refer to "Bringing Up a Switch Mesh Domain" on page 5-8.)
- To view the current switch mesh status on the switch, use the CLI **show mesh** command (page 5-12).

Menu: To Configure Switch Meshing

1. From the Main Menu, select:

   2. Switch Configuration
   2. Port/Trunk Settings

2. Press **[E]** (for **Edit**) to access the load balancing parameters.

```
============================ CONSOLE - MANAGER MODE ============================
                    Switch Configuration - Port/Trunk Settings

  Port     Type       Enabled      Mode       Flow Ctrl  Group  Type
  ----  ---------  + -------  ------------  ---------  -----  -----
  A1    1000SX     | Yes      Auto          Disable
  A2    1000SX     | Yes      Auto          Disable
  A3    1000LX     | Yes      Auto          Disable
  A4    1000LX     | Yes      Auto          Disable
  B1    1000T      | Yes      Auto          Disable
  B2    1000T      | Yes      Auto          Disable
  B3    1000T      | Yes      Auto          Disable
  B4    1000T      | Yes      Auto          Disable
  C1    10/100TX   | Yes      Auto          Disable
  C2    10/100TX   | Yes      Auto          Disable
  C3    10/100TX   | Yes      Auto          Disable
  C4    10/100TX   | Yes      Auto          Disable

 Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 5-6. Example of the Screen for Configuring Ports for Meshing

3. In the Group column, move the cursor to the port you want to assign to the switch mesh.
4. Press **[M]** to choose **Mesh** for the selected port.
5. Use the **up-arrow or down-arrow** key to select the next port you want to include in your mesh domain, then press **[M]** again. For example, if you were adding ports A1 and A2 to your mesh domain, the screen would appear similar to figure 5-7:

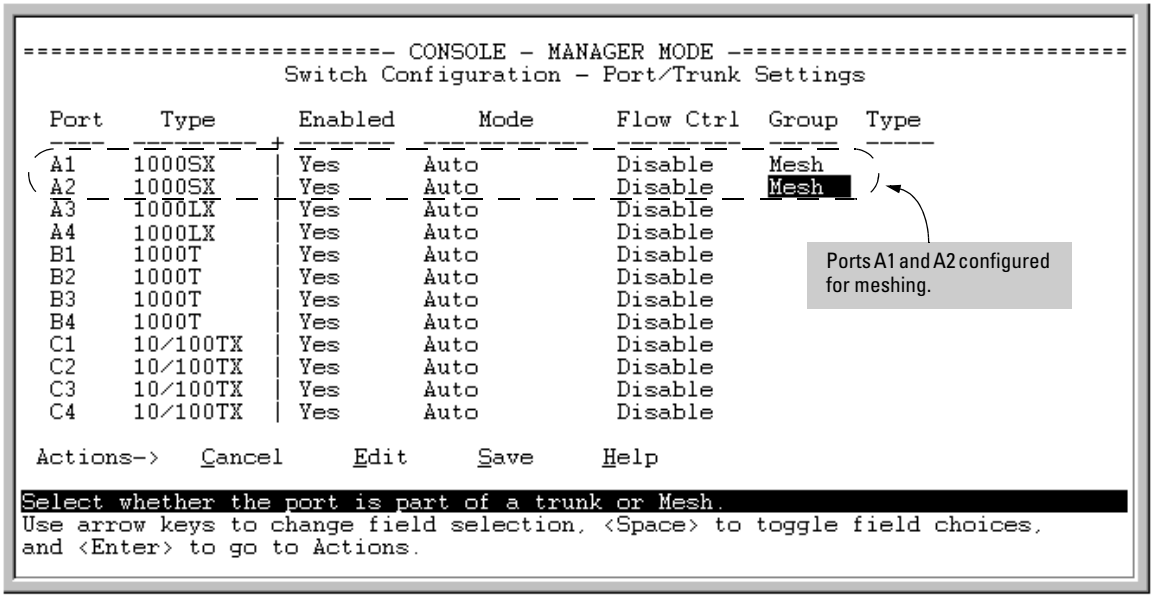

CONSOLE - MANAGER MODE
Switch Configuration - Port/Trunk Settings
Port Type Enabled Mode Flow Ctrl Group Type
A1 1000SX Yes Auto Disable Mesh
A2 1000SX Yes Auto Disable Mesh
A3 1000LX Yes Auto Disable
A4 1000LX Yes Auto Disable
B1 1000T Yes Auto Disable
B2 1000T Yes Auto Disable
B3 1000T Yes Auto Disable
B4 1000T Yes Auto Disable
C1 10/100TX Yes Auto Disable
C2 10/100TX Yes Auto Disable
C3 10/100TX Yes Auto Disable
C4 10/100TX Yes Auto Disable
Actions-> Cancel Edit Save Help
Select whether the port is part of a trunk or Mesh.
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
Ports A1 and A2 configured for meshing.

Figure 5-7. Example of Mesh Group Assignments for Several Ports

6. Repeat step 5 for all ports you want in the mesh domain.

Notes

For meshed ports, leave the **Type** setting blank. (Meshed ports do not accept a **Type** setting.)

All meshed ports in the switch automatically belong to the same mesh domain. (See figure 5-2 on page 5-4.)

7. When you finish assigning ports to the switch mesh, press **[Enter],** then **[S]** (for **Save**). You will then see the following screen.

The asterisk indicates that you must reboot the switch to cause the **Mesh** configuration change to take effect.

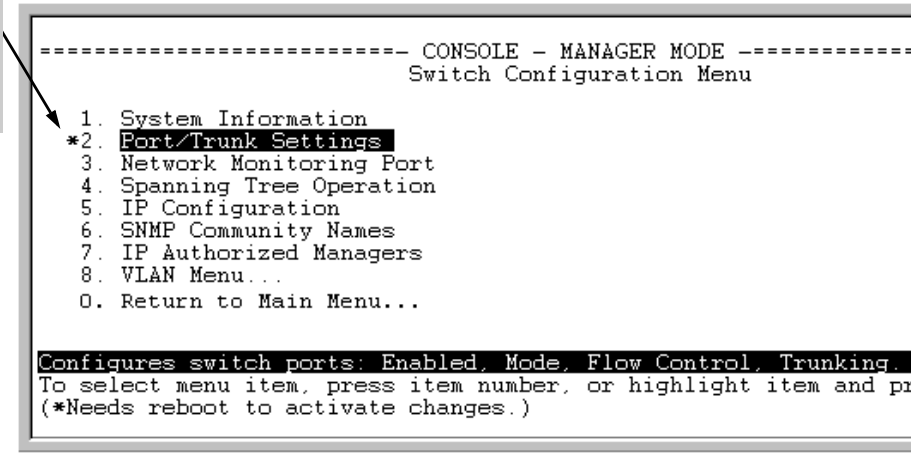

========== CONSOLE - MANAGER MODE ==========
Switch Configuration Menu
1. System Information
*2. Port/Trunk Settings
3. Network Monitoring Port
4. Spanning Tree Operation
5. IP Configuration
6. SNMP Community Names
7. IP Authorized Managers
8. VLAN Menu...
0. Return to Main Menu...
Configures switch ports: Enabled, Mode, Flow Control, Trunking.
To select menu item, press item number, or highlight item and pr
(*Needs reboot to activate changes.)

Figure 5-8. After Saving a Mesh Configuration Change, Reboot the Switch

8. Press **[0]** to return to the Main menu.
9. To activate the mesh assignment(s) from the Main menu, reboot the switch by pressing the following keys:
   a. **[6]** (for **Reboot Switch**)
   b. Space bar (to select **Yes**).
   c. **13** (to start the reboot process).

(The switch cannot dynamically reconfigure ports to enable or disable meshing, so it is always necessary to reboot the switch after adding or deleting a port in the switch mesh.)

CLI: To View and Configure Switch Meshing

Port Status and Configuration Features

Feature	Default	Menu	CLI	Web
viewing switch mesh status	n/a	n/a	below	n/a
configuring switch meshing	Disabled	n/a		n/a

Viewing Switch Mesh Status

Syntax: show mesh

Lists the switch ports configured for meshing, along with the **State** *of each mesh-configured connection, the MAC address of the switch on the opposite end of the link (***Adjacent Switch***), and the MAC address of the port on the opposite end of the link (***Peer Port***).*

Reading the Show Mesh Output. For each port configured for meshing, the State column indicates whether the port has an active link to the mesh or is experiencing a problem.

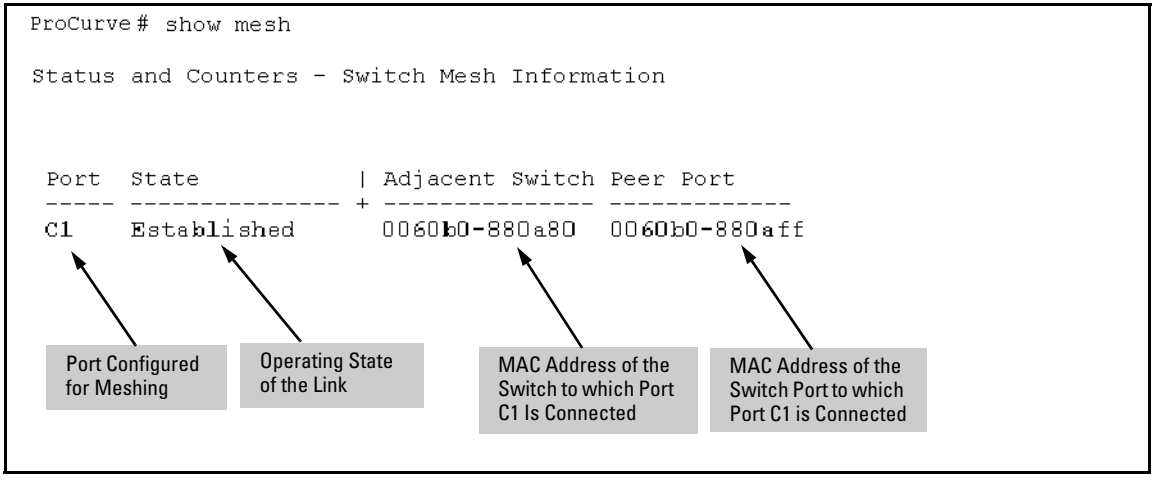

ProCurve # show mesh
Status and Counters - Switch Mesh Information
Port State | Adjacent Switch Peer Port
C1 Established 0060b0-880a80 0060b0-880aff
Port Configured for Meshing
Operating State of the Link
MAC Address of the Switch to which Port C1 Is Connected
MAC Address of the Switch Port to which Port C1 is Connected

Figure 5-9. Example of the Show Mesh Report

Table 5-1. State Descriptions for Show Mesh Output

State	Meaning
Established	The port is linked to a meshed port on another switch and meshing traffic is flowing across the link. The **show mesh** listing includes the MAC addresses of the adjacent switch and direct connection port on the adjacent switch.
Not Established	The port may be linked to a switch on a port that is not configured for meshing or has gone down.
Initial	The port has just come up as a meshed port and is trying to negotiate meshing.
Disabled	The port is configured for meshing but is not connected to another device.
Error	Indicates a multiple MAC-address error. This occurs when you have two or more mesh ports from the same switch linked together through a hub.
Topology Error	Two meshed switches are connected via a hub, and traffic from other, non-meshed devices, is flowing into the hub. The **show mesh** listing includes the MAC addresses of the adjacent switch and direct connection port on the adjacent switch.

Topology Example with Show Mesh. Suppose that you have the following topology:

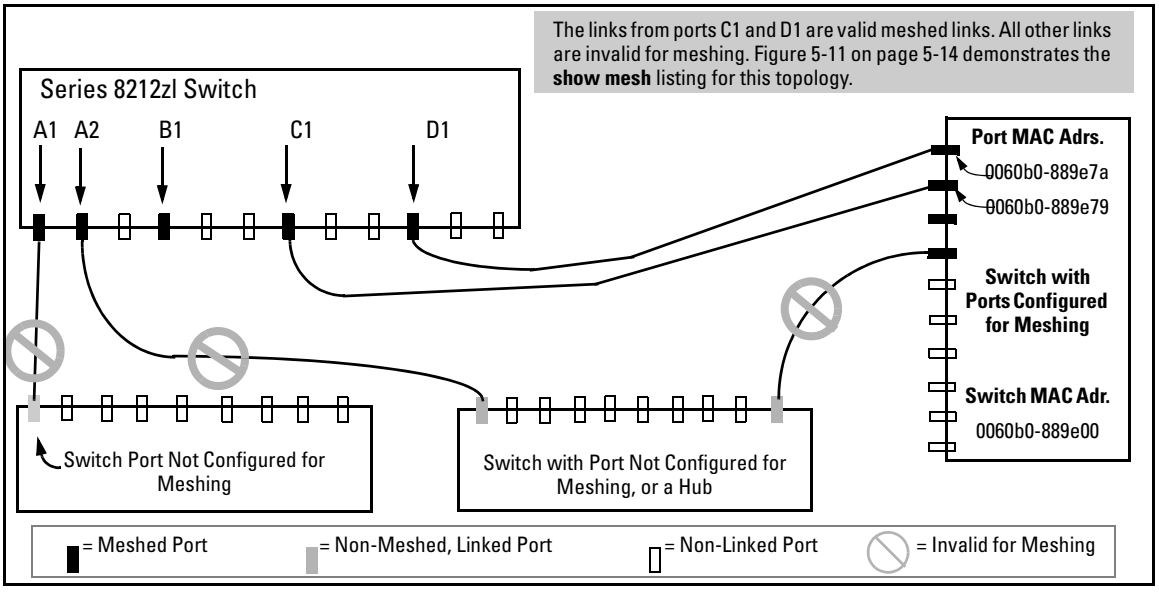

The links from ports C1 and D1 are valid meshed links. All other links are invalid for meshing. Figure 5-11 on page 5-14 demonstrates the show mesh listing for this topology.
Series 8212zl Switch
A1
A2
B1
C1
D1
Port MAC Adrs.
0060b0-889e7a
0060b0-889e79
Switch with Ports Configured for Meshing
Switch MAC Adr.
0060b0-889e00
Switch Port Not Configured for Meshing
Switch with Port Not Configured for Meshing, or a Hub
= Meshed Port
= Non-Meshed, Linked Port
= Non-Linked Port
= Invalid for Meshing

Figure 5-10. Example of a Meshed Topology with Some Mesh Ports Incorrectly Linked

Table 5-2 on page 5-14 describes the meshing operation in the above topology.

Table 5-2. Operating Details for Figure 5-10

Port	Meshing?	Connection
A1	Yes	Connected to a port that may not be configured for meshing
A2	Yes	Connected to a switch port on a device that is not configured for meshing (another switch, or a hub). In this case, the **Topology Error** message indicates that the switch detects a meshed port on another, non-adjacent device that is also connected to the non-meshed switch or hub. ***However, meshing will not operate properly through this connection.***
B1	Yes	Not connected to another device.
C1	Yes	Connected to a meshed port on the same adjacent switch as D1 with meshing operating properly.
D1	Yes	Connected to a meshed port on the same adjacent switch as C1 with meshing operating properly.

Figure 5-11 lists the show mesh display for the topology and meshing configuration in figure 5-10:

```
ProCurve# show mesh

Status and Counters - Switch Mesh Information


 Port  State            | Adjacent Switch  Peer Port
 ----- ---------------- + ---------------- --------------
 A1    Not Established
 A2    Topology Error     0060b0-889e00    0060b0-889e7b
 B1    Disabled
 C1    Established        0060b0-889e00    0060b0-889e7a
 D1    Established        0060b0-889e00    0060b0-889e79
```

Figure 5-11. Example of the Show Mesh Listing for the Topology in Figure 5-10

CLI: Configuring Switch Meshing

Syntax: [no] mesh [e] < *port-list* >

Enables or disables meshing operation on the specified ports.

All meshed ports on a switch belong to the same mesh domain. Thus, to configure multiple meshed ports on a switch, you need to:

1. Specify the ports you want to operate in the mesh domain.
2. Use **write memory** to save the configuration to the startup-config file.
3. Reboot the switch

For example, to configure meshing on ports A1-A4, B3, C1, and D1-D3:

```
ProCurve (config)# mesh e a1-a4,b3,c1,d1-d3
Command will take effect after saving configuration and reboot.
ProCurve (config)# write memory
ProCurve (config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 5-12. Example of How To Configure Ports for Meshing

To remove a port from meshing, use the "**no**" version of **mesh**, followed by **write memory** and rebooting the switch. For example, to remove port C1 from the mesh:

```
ProCurve # config
ProCurve (config)# no mesh c1
Command will take effect after saving configuration and reboot.
ProCurve (config)# write memory
ProCurve (config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 5-13. Example of Removing a Port from the Mesh

Operating Notes for Switch Meshing

In a switch mesh domain traffic is distributed across the available paths with an effort to keep latency the same from path to path. The path selected at any time for a connection between a source node and a destination node is based on these latency and throughput cost factors:

- Outbound queue depth, or the current outbound load factor for any given outbound port in a possible path
- Port speed, such as 10Mbps versus 100Mbps; full-duplex or half-duplex
- Inbound queue depth, or how busy a destination switch is in a possible path
- Increased packet drops, indicating an overloaded port or switch

Paths having a lower cost will have more traffic added than those having a higher cost. Alternate paths and cost information is discovered periodically and communicated to the switches in the mesh domain. This information is used to assign traffic paths between devices that are newly active on the mesh.

This means that after an assigned path between two devices has timed out, new traffic between the same two devices may take a different path than previously used.

To display information on the operating states of meshed ports and the identities of adjacent meshed ports and switches, see "Viewing Switch Mesh Status" on page 5-12.

Flooded Traffic

Broadcast and multicast packets will always use the same path between the source and destination edge switches unless link failures create the need to select new paths. (Broadcast and multicast traffic entering the mesh from different edge switches are likely to take different paths.) When an edge switch receives a broadcast from a non-mesh port, it floods the broadcast out all its other non-mesh ports, but sends the broadcast out only those ports in the mesh that represent the path from that edge switch through the mesh domain. (Only one copy of the broadcast packet gets to each edge switch for broadcast out of its nonmeshed ports. This helps to keep the latency for these packets to each switch as low as possible.)

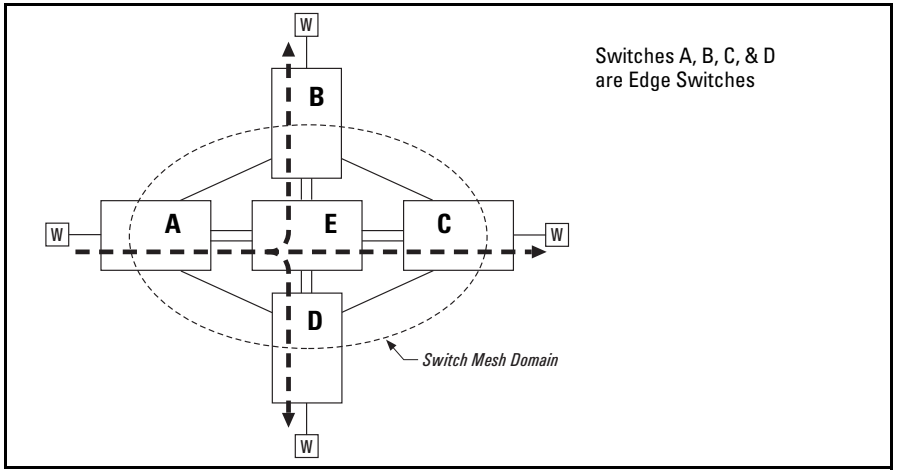


Figure 5-14. Example of a Broadcast Path Through a Switch Mesh Domain

Any mesh switches that are not edge switches will flood the broadcast packets only through ports (paths) that link to separate edge switches in the controlled broadcast tree. The edge switches that receive the broadcast will flood the broadcast out all non-meshed ports. Some variations on broadcast/multicast

traffic patterns, including the situation where multiple VLANs are configured and a broadcast path through the mesh domain leads only to ports that are in the same VLAN as the device originating the broadcast.

Unicast Packets with Unknown Destinations

A meshed switch receiving a unicast packet with an unknown destination does not flood the packet onto the mesh. Instead, the switch sends a query on the mesh to learn the location of the unicast destination. The meshed switches then send 802.2 test packets through their non-meshed ports. After the unicast destination is found and learned by the mesh, subsequent packets having the same destination address will be forwarded. By increasing the **MAC Age Time** you can cause the switch address table to retain device addresses longer. (For more on **MAC Age Time**, refer to "System Information" in the chapter titled "Interface Access and System Information" in the *Management and Configuration Guide* for your switch.) Because the switches in a mesh exchange address information, this will help to decrease the number of unicast packets with unknown destinations, which improves latency within the switch mesh. Also, in an IP environment, ProCurve recommends that you configure IP addresses on meshed switches. This makes the discovery mechanism more robust, which contributes to decreased latency.

Spanning Tree Operation with Switch Meshing

Using MSTP with several switches and no switch meshing configured can result in unnecessarily blocking links and reducing available bandwidth. For example:

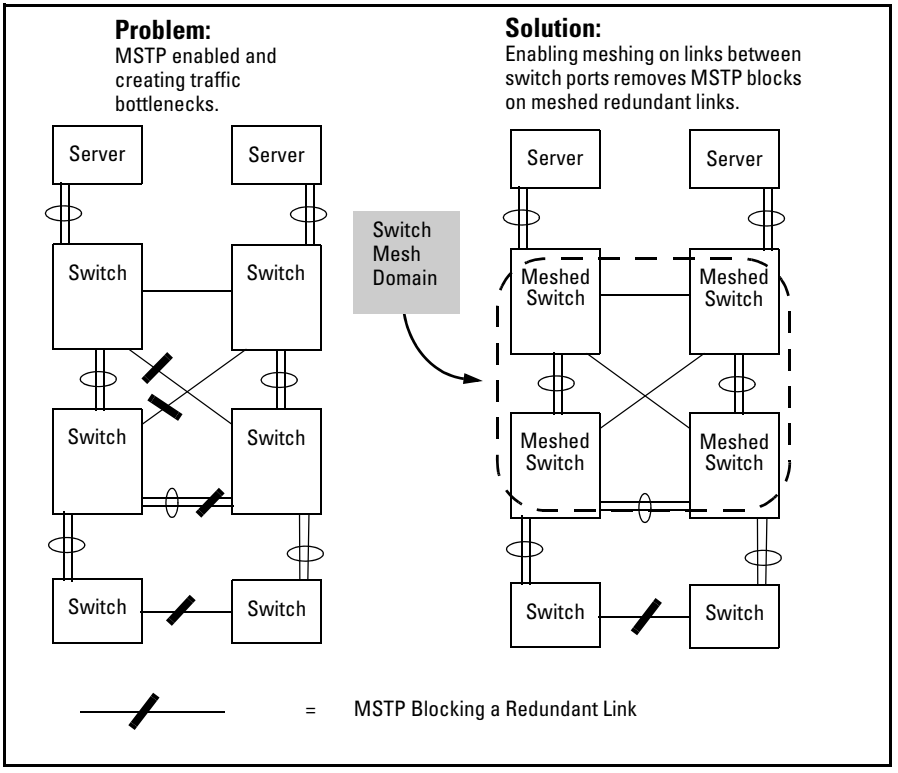


Figure 5-15. Example Using STP Without and With Switch Meshing

If you are going to use spanning-tree in a switch mesh, all switches in the mesh should be configured with the same type of spanning-tree: 802.1d/STP, 802.1w/RSTP, or 802.1s/MSTP. Spanning-Tree interprets a meshed domain as a single link. However, on edge switches in the domain, MSTP will manage non-meshed redundant links from other devices. For example:

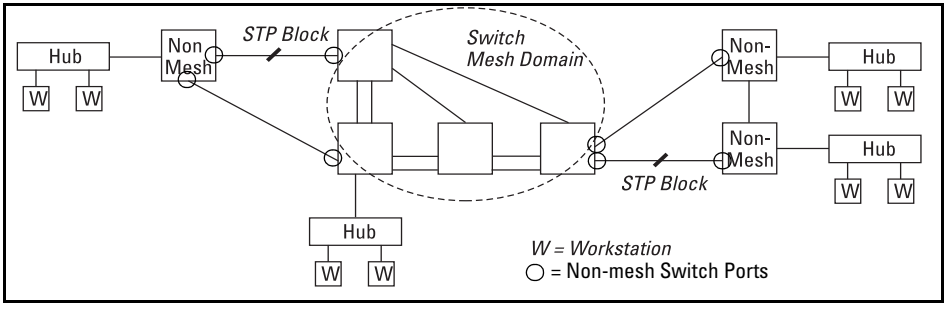


Figure 5-16. Connecting a Switch Mesh Domain to Non-Meshed Devices

Note on the Edge-Port Mode in MSTP

When using MSTP and interconnecting switches covered in this guide in a mesh with switches that are not in the mesh, all the non-mesh switch ports (as indicated in the figure above) should have the **edge-port** parameter disabled.

MSTP should be configured on non-mesh devices that use redundant links to interconnect with other devices or with multiple switch mesh domains. For example:

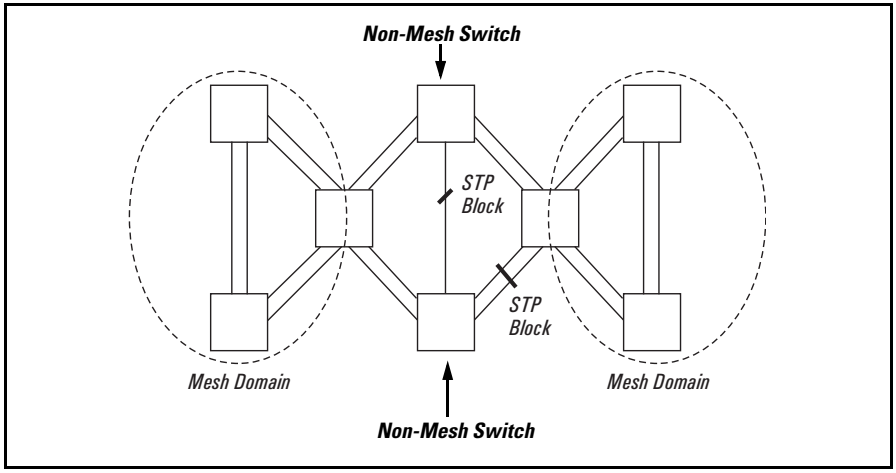


Figure 5-17. Interconnecting Switch Mesh Domains with Redundant Links

In the above case of multiple switch meshes linked with redundant trunks there is the possibility that spanning-tree will temporarily block a mesh link. This is because it is possible for spanning-tree to interpret the cost on an external trunked link to be less than the cost on a meshed link. However, if this condition occurs, the meshed switch that has a blocked link will automatically increase the cost on the external (non-meshed) link to the point where spanning tree will block the external link and unblock the meshed link. This process typically resolves itself in approximately 30 seconds.

Caution

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default spanning-tree parameter settings are usually adequate for spanning tree operation. Also, because incorrect spanning tree settings can adversely affect network performance, you should not make changes unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default MSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the MSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For more on spanning-tree, refer to chapter 4, "Multiple Instance Spanning-Tree Operation" in this guide. Also, you may want to examine the IEEE 802.1d, 802.1w, or 802.1s standards, depending on which version of spanning-tree you are using. The switches covered in this guide use 802.1s.

Filtering/Security in Meshed Switches

Because paths through the mesh can vary with network conditions, configuring filters on meshed ports can create traffic problems that are difficult to predict, and is not recommended. However, configuring filters on nonmeshed ports in an edge switch provides you with control and predictability.

IP Multicast (IGMP) in Meshed Switches

Like trunked ports, the switch mesh domain appears as a single port to IGMP. However, unlike trunked ports, IGMP protocol and multicast traffic may be sent out over several links in the mesh in the same manner as broadcast packets.

Static VLANs

In a network having a switch mesh domain and multiple static VLANs configured, all static VLANs must be configured on each meshed switch, even if no ports on the switch are assigned to any VLAN. (The switch mesh is a member of all static VLANs configured on the switches in the mesh.)

When static VLANs are configured, the mesh is seen as a single entity by each VLAN. All ports in the mesh domain are members of all VLANs and can be used to forward traffic for any VLAN. However, the non-mesh ports on edge switches that allow traffic to move between the mesh and non-meshed devices belong to specific VLANs and do not allow packets originating in a specific VLAN to enter non-meshed devices that do not belong to that same VLAN. (It is necessary to use a router to communicate between VLANs.) For example, in the following illustration, traffic from host A entering the switch mesh can only exit the mesh at the port for hosts B and E. Traffic from host A for any other host (such as C or D) will be dropped because only hosts B and E are in the same VLAN as host A.

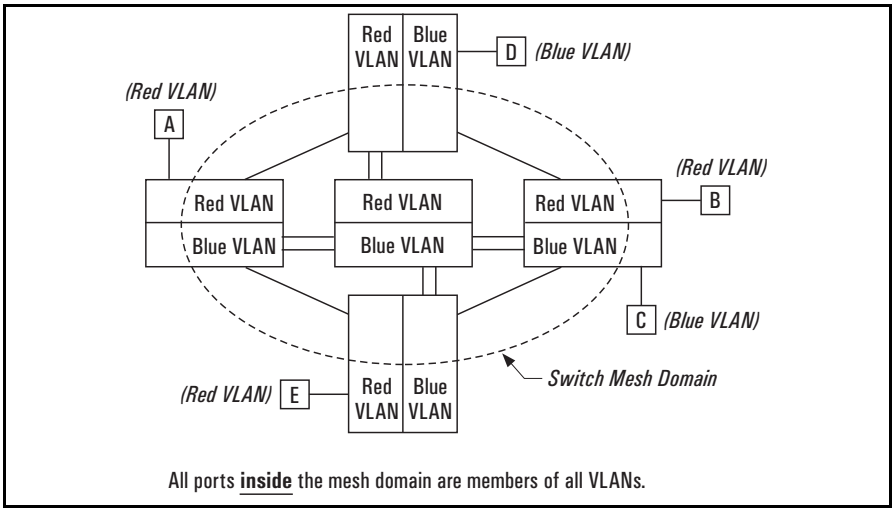


Figure 5-18. VLAN Operation with a Switch Mesh Domain

Dynamic VLANs

If GVRP is enabled, meshed ports in a switch become members of any dynamic VLANs created in the switch in the same way that they would if meshing was not configured in the switch. (For more on GVRP, refer to chapter 3, "GVRP".)

Jumbo Packets

If you enable jumbo traffic on any VLAN, then all meshed ports on the switch will be enabled to support jumbo traffic. (On a given meshed switch, every meshed port becomes a member of every VLAN configured on the switch.) If a port in a meshed domain does not belong to any VLANs configured to support jumbo traffic, then the port drops any jumbo packets it receives from other devices. In this regard, if a mesh domain includes any ProCurve 8212zl switches, 6200yl switches, Series 5400zl switches, Series 3500yl switches, Series 3400cl or Series 6400cl switches that are configured to support jumbo traffic, only these switches can transmit and receive jumbo packets. Other switch models in the mesh will drop jumbo packets as they are not supported by those switches. For more information on jumbo packets, refer to the chapter titled "Port Traffic Controls" in the *Management and Configuration Guide* for your switch.

Mesh Design Optimization

Mesh performance can be enhanced by using mesh designs that are as small and compact as possible while still meeting the network design requirements. The following are limits on the design of meshes and have not changed:

1. Any switch in the mesh can have up to 24 meshed ports.
2. A mesh domain can contain up to 12 switches.
3. Up to 5 inter-switch meshed hops are allowed in the path connecting two nodes.
4. A fully interconnected mesh domain can contain up to 5 switches.

Mesh performance can be optimized by keeping the number of switches and the number of possible paths between any two nodes as small as possible. As mesh complexity grows, the overhead associated with dynamically calculating and updating the cost of all of the possible paths between nodes grows exponentially. Cost discovery packets are sent out by each switch in the mesh every 30 seconds and are flooded to all mesh ports. Return packets include a cost metric based on inbound and outbound queue depth, port speed, number of dropped packets, etc. Also, as mesh complexity grows, the number of hops over which a downed link has to be reported may increase, thereby increasing the reconvergence time.

The simplest design is the two-tier design because the number of possible paths between any two nodes is kept low and any bad link would have to be communicated only to it's neighbor switch.

Other factors affecting the performance of mesh networks include the number of destination addresses that have to be maintained, and the overall traffic levels and patterns. However a conservative approach when designing new mesh implementations is to use the two-tier design and limit the mesh domain to eight switches where possible.

As shown here, meshing allows multiple redundant links between switches in the domain, and more than one link is allowed between any two switches in the domain.

Note also that a switch can have up to 24 ports configured for meshing.

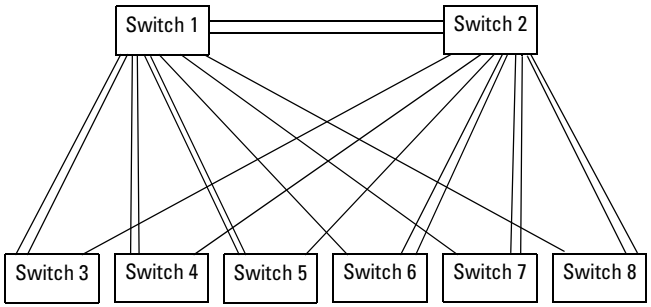
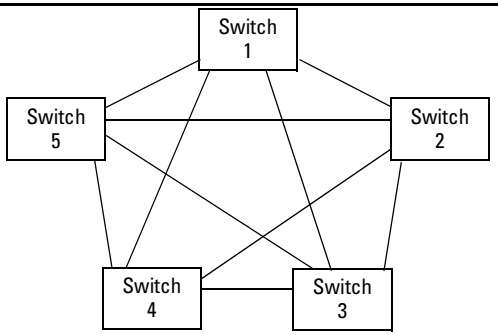


Figure 5-19. Example of a Two-Tier Mesh Design



As noted above, a switch can have up to 24 ports configured for meshing.

Figure 5-20. Example of a Fully Interconnected Mesh with the Maximum Switch Count

Other factors affecting the performance of mesh networks include the number of destination addresses that have to be maintained, and the overall traffic levels and patterns. However a conservative approach when designing new mesh implementations is to use the two-tier design and limit the mesh domain to eight switches where possible.

Other Requirements and Restrictions

- **Mesh Support Within the Domain:** All switches in the mesh domain, including edge switches, must support the ProCurve switch meshing protocol.
- **Switch Hop Count in the Mesh Domain:** A maximum of five (meshed) switch hops is allowed in the path connecting two nodes in a switch mesh domain. A path of six meshed hops is unusable. However, this does not interfere with other, shorter paths in the same domain.

- **Connecting Mesh Domains:** To connect two separate switch meshing domains, you must use non-meshed ports. (The non-meshed link can be a port trunk or a single link.) Refer to figure 5-3 on page 5-6.
- **Multiple Links Between Meshed Switches:** Multiple mesh ports can be connected between the same two switches, to provide higher bandwidth. Each port that you want in the mesh domain should be configured as **Mesh** (and not as a trunk—**Trk**). Note that if you configure a port as **Mesh**, there is no "Type" selection for that port.
- **Network Monitor Port:** If a network monitor port is configured, broadcast packets may be duplicated on this port if more than one port is being monitored and switch meshing is enabled.
- **Compatibility with Other Switches:** The switches covered in this guide operate with the Series 5300xl switches in normal mode.
- **Rate-Limiting Not Recommended on Meshed Ports: Rate-Limiting can reduce the efficiency of paths through a mesh domain.**

(See also "Operating Rules" on page 5-5.)

For additional information on troubleshooting meshing problems, refer to "Using a Heterogeneous Switch Mesh" on page 5-7 and "Mesh-Related Problems" in appendix C, "Troubleshooting" of the Management and Configuration Guide for your switch.

6

Quality of Service: Managing Bandwidth More Effectively

Contents

Using Quality of Service Policies 6-4

QoS Terminology 6-7

QoS Operation 6-9

Globally-Configured QoS 6-10

Classifier-Based QoS 6-11

QoS Packet Classification 6-12

Globally-Configured Packet Classification 6-12

Classifier-Based Match Criteria 6-13

QoS Traffic Marking 6-14

Globally-Configured Traffic Marking 6-14

Layer 2 802.1p Prioritization 6-14

Layer 3 DSCP Marking 6-16

VLAN and Untagged VLAN Environments 6-17

Classifier-Based Traffic Marking 6-18

Globally-Configured QoS 6-19

Global QoS Configuration Procedure 6-19

Viewing a Global QoS Configuration 6-21

Global QoS Restrictions 6-22

Global TCP/UDP Classifier 6-24

Assigning an 802.1p Priority for a Global TCP/UDP Classifier . 6-25

Operating Notes on Using TCP/UDP Port Ranges 6-26

Assigning a DSCP Policy for a Global TCP/UDP Classifier 6-27

Displaying Resource Usage for QoS Policies 6-32

Global IP-Device Classifier 6-33

Assigning a Priority for a Global IP-Device Classifier 6-34

Assigning a DSCP Policy For a Global IP-Device Classifier 6-36

Global IP Type-of-Service Classifier 6-41
IPv4 ToS/IPv6 Traffic Class Byte 6-42
Assigning an 802.1p Priority for a Global IP-Precedence Classifier 6-44
Assigning an 802.1p Priority for a Global IP-Diffserv Classifier 6-45
Assigning a DSCP Policy for a Global IP-Diffserv Classifier . . . 6-49
Comparison of Global IP Type-of-Service Classifiers 6-53
Global Layer-3 Protocol Classifier 6-54
Assigning a Priority for a Global Layer-3 Protocol Classifier . . . 6-54
Global VLAN-ID Classifier 6-56
Assigning a Priority for a Global VLAN-ID Classifier 6-56
Assigning a DSCP Policy for a Global VLAN-ID Classifier 6-58
Global Source-Port Classifier 6-62
Assigning a Priority for a Global Source-Port Classifier 6-62
Assigning a DSCP Policy for a Global Source-Port Classifier . . 6-64
IP Multicast (IGMP) Interaction with QoS 6-69

Advanced Classifier-Based QoS 6-70
Classifier-Based QoS Model 6-71
Classifier-Based QoS Configuration Procedure 6-71
Configuring QoS Actions in a Policy 6-76
Override of Global QoS Settings 6-81
Viewing a Classifier-Based QoS Configuration 6-81
Classifier-Based QoS Restrictions 6-85
Interaction with Other Software Features 6-85
Classifier-Based QoS Configuration Examples 6-86
QoS Policy for Layer 4 TCP/UDP Traffic 6-87
QoS Policy for Subnet Traffic 6-87

Differentiated Services Codepoint (DSCP) Mapping 6-88
Default Priority Settings for Selected Codepoints 6-89
Displaying Non-Default Codepoint Settings 6-90
Notes on Changing a Priority Setting 6-91
Error Messages for DSCP Policy Changes 6-92
Example of Changing the Priority Setting on a Policy When One or More Classifiers Are Currently Using the Policy . 6-92

QoS Queue Configuration 6-95

Configuring the Number of Priority Queues 6-96
Viewing the QoS Queue Configuration 6-98

Using Quality of Service Policies

A Quality of Service (QoS) *network policy* refers to the network-wide controls you can implement to:

- Ensure uniform and efficient traffic-handling throughout your network, while keeping the most important traffic moving at an acceptable speed, regardless of current bandwidth usage.
- Exercise control over the priority settings of inbound traffic arriving in and travelling through your network.

Adding bandwidth is often a good idea, but it is not always feasible and does not completely eliminate the potential for network congestion. There will always be points in the network where multiple traffic streams merge or where network links change speed and capacity. The impact and number of these congestion points will increase over time as more applications and devices are added to the network.

When (not *if*) network congestion occurs, it is important to move traffic on the basis of relative importance. However, without *Quality of Service* (QoS) prioritization, less important traffic can consume network bandwidth and slow down or halt the delivery of more important traffic. That is, without QoS, most traffic received by the switch is forwarded with the same priority it had upon entering the switch. In many cases, such traffic is "normal" priority and competes for bandwidth with all other normal-priority traffic, regardless of its relative importance to your organization's mission.

Use QoS to classify and prioritize network traffic. Quality of Service is used to classify and prioritize traffic throughout a network. QoS enables you to establish an end-to-end traffic-priority policy to improve the control and throughput of important data. You can manage available bandwidth so that the most important traffic goes first. For example, you can use Quality of Service to:

- Upgrade or downgrade traffic from various servers.
- Control the priority of traffic from dedicated VLANs or applications.
- Change the priorities of traffic from various segments of your network as your business needs change.
- Set priority policies in edge switches in your network to enable traffic-handling rules across the network.

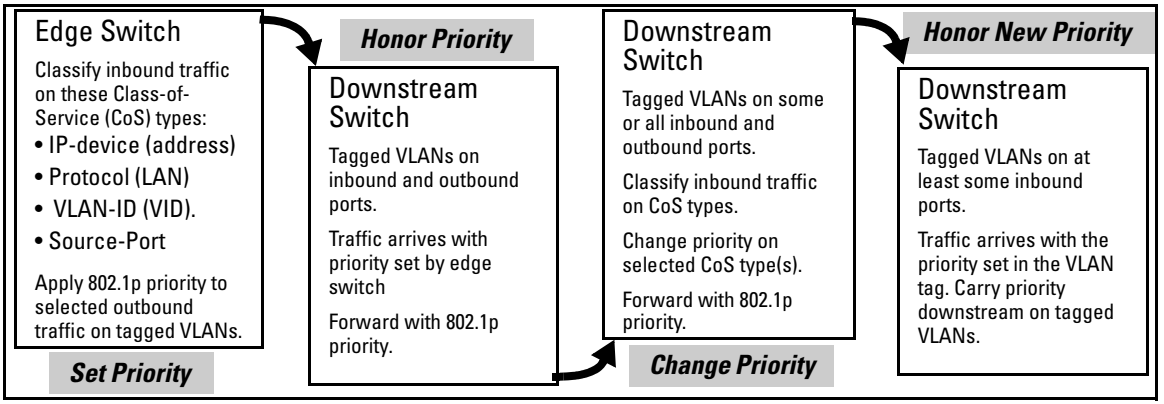

Edge Switch
Classify inbound traffic on these Class-of-Service (CoS) types:
• IP-device (address)
• Protocol (LAN)
• VLAN-ID (VID).
• Source-Port
Apply 802.1p priority to selected outbound traffic on tagged VLANs.
Set Priority
Honor Priority
Downstream Switch
Tagged VLANs on inbound and outbound ports.
Traffic arrives with priority set by edge switch
Forward with 802.1p priority.
Downstream Switch
Tagged VLANs on some or all inbound and outbound ports.
Classify inbound traffic on CoS types.
Change priority on selected CoS type(s).
Forward with 802.1p priority.
Change Priority
Honor New Priority
Downstream Switch
Tagged VLANs on at least some inbound ports.
Traffic arrives with the priority set in the VLAN tag. Carry priority downstream on tagged VLANs.

Figure 6-1. Example of 802.1p Priority Based on CoS (Class-of-Service) Types and Use of VLAN Tags

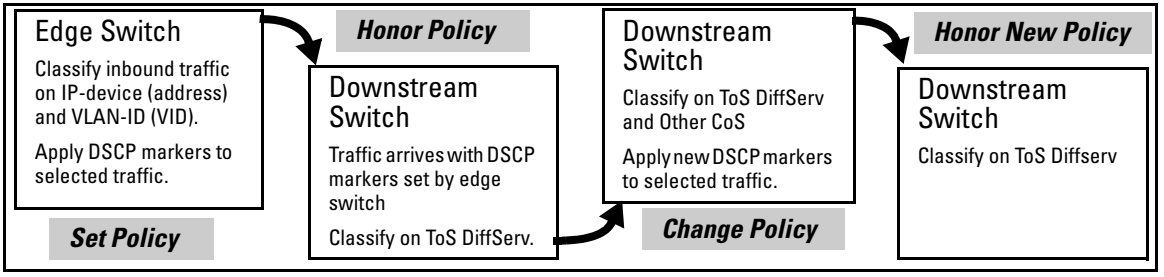

Edge Switch
Classify inbound traffic on IP-device (address) and VLAN-ID (VID).
Apply DSCP markers to selected traffic.
Set Policy
Honor Policy
Downstream Switch
Traffic arrives with DSCP markers set by edge switch
Classify on ToS DiffServ.
Downstream Switch
Classify on ToS DiffServ and Other CoS
Apply new DSCP markers to selected traffic.
Change Policy
Honor New Policy
Downstream Switch
Classify on ToS Diffserv

Figure 6-2. Example Application of Differentiated Services Codepoint (DSCP) Policies

Apply QoS to inbound traffic at the network edge. At the edge switch, QoS classifies certain traffic types and in some cases applies a DSCP policy. At the next hop (downstream switch) QoS honors the policies established at the edge switch. Further downstream, another switch may reclassify some traffic by applying new policies, and yet other downstream switches can be configured to honor the new policies.

Preserve QoS in outbound traffic in a VLAN. QoS is implemented in the form of rules or policies that are configured on the switch. Although you can use QoS to prioritize traffic only while it moves through the switch, you derive the *maximum benefit* by using QoS in an 802.1Q VLAN environment (with 802.1p priority tags) or in an untagged VLAN environment (with DSCP policies in which QoS sets priorities that downstream devices can support without re-classifying the traffic).

Use QoS to optimize existing network resources. By prioritizing traffic, QoS supports traffic growth on the network while optimizing the use of existing resources—and delaying the need for further investments in equipment and services. QoS enables you to:

- Specify which traffic has higher or lower priority, regardless of current network bandwidth or the relative priority setting of the traffic when it is received on the switch.
- Change (upgrade or downgrade) the priority of outbound traffic.
- Override "illegal" packet priorities set by upstream devices or applications that use 802.1Q VLAN tagging with 802.1p priority tags.

Use classifier-based QoS to provide additional policy actions and aid migration in networks with legacy and OEM devices.
Starting in software release K.14.01, ProCurve QoS configuration supports a classifier-based model that provides added functionality to create and manage QoS policies across a network consisting of ProCurve switches as well as OEM and legacy devices.

The classifier-based configuration model is a single, simplified procedure and command syntax for cross-feature usage, which offers:

- Finer granularity than globally-configured QoS for classifying IPv4 and IPv6 traffic
- Additional actions for managing selected traffic, such as rate-limiting and IP precedence marking
- The application of QoS policies to inbound traffic flows on specific port and VLAN interfaces (instead of using only globally-configured, switch-wide QoS settings)
- The use of configured traffic classes by different software features, such as QoS or port mirroring

Classifier-based QoS is designed to work with existing globally-configured, switch-wide QoS policies by allowing you to zoom in on a subset of port or VLAN traffic to further manage it. Classifier-based policies take precedence over and may override globally-configured, switch-wide QoS settings.

Classifier-based QoS policies provide greater control for managing network traffic. Using multiple match criteria, you can finely select and define the classes of traffic that you want to manage. QoS-specific actions determine how you can handle the selected traffic.

QoS Terminology

Term	Use in This Document
802.1p priority	A traffic priority setting carried by a VLAN-tagged packet moving from one device to another through ports that are tagged members of the VLAN to which the packet belongs. An 802.1p priority can be from 0 to 7. The switch handles an outbound packet on the basis of its 802.1p priority, using the priority to assign the packet to the appropriate outbound port queue. If a packet leaves the switch on a VLAN port that is an untagged member of the VLAN, the priority is dropped and the packet arrives at the next, downstream device without an 802.1p priority assignment.
802.1Q field	A four-byte field contained in the header of Ethernet packets entering or leaving the switch through a port that is a tagged member of a VLAN. This field includes an 802.1p priority setting, a VLAN tag, or ID number (VID), and other data. A packet entering or leaving the switch through a port that is an untagged member of the outbound VLAN does not have this field in its header and thus does not carry a VID or an 802.1p priority. See also "802.1p priority".
class	Defines a set of different types of traffic (using **match** and **ignore** commands) to be managed in a specific way (for example, prioritizing all High Availability traffic).
classifier	Method of matching or ignoring certain traffic attributes that allows you to classify packets using more than one traffic attribute at a time. This method can be used by multiple features, such as QoS and mirroring.
classifier model	Applied to various software features, this configuration model requires you to first classify the traffic that you want to manage, and then configure a policy containing the actions to be executed on the class.
codepoint	See DSCP, below.
CoS	***Class of Service.*** Priority level (0-7) used to transmit a packet on an outbound queue. For an individual packet, class of service is determined by the Layer 2 802.1p priority value in a packet header. The 802.1p priority is associated with a Layer 3 IP precedence bit-value or DSCP codepoint (see Table 6-11).
Differentiated Services (Diff-Serv) mode	See Type-of-Service (ToS), below.
downstream device	A device linked directly or indirectly to an outbound switch port. The switch sends traffic to downstream devices.
DSCP	***Differentiated Services Codepoint.*** (Also known as ***codepoint***.) A DSCP consists of the upper six bits of the: • Type of Service (ToS) byte in an IPv4 packet • Traffic Class byte in an IPv6 packet • There are 64 possible DSCP codepoints. In the default QoS configuration for the switches covered in this guide, some codepoints (such as Assured Forwarding and Expedited Forwarding) are configured by default with an 802.1p priority. All other codepoints have no 802.1p priority assigned and are listed as **No-override** (see Table 6-11).

Term	Use in This Document
DSCP policy	A DSCP codepoint that is configured with an 802.1p priority (0 to 7). Default: **No-override**. Using a DSCP policy, you can configure the switch to prioritize IP packets that match a specified classifier by assigning a new DSCP and 802.1p priority (0-7). For more information on DSCP, refer to "IPv4 ToS/IPv6 Traffic Class Byte" on page 6-42. For an example of the DSCP bit set in a ToS or Traffic Class field, see Figure 6-13.
edge switch	In QoS, an edge switch is a switch that receives traffic from the edge of the LAN or from outside the LAN, and forwards it to devices within the LAN. Typically, an edge switch is used with QoS to identify packets based on classifiers such as TCP/UDP application type, IP-device (address), Layer 3 protocol (LAN), VLAN-ID (VID), and source port (although it can also be used to classify packets on the basis of IP precedence and DSCP bits). Using this packet recognition, the edge switch can be used to set 802.1p priorities or DSCP policies that downstream devices will honor.
inbound port	Any port on the switch through which traffic enters the switch.
IP options	Optional fields supported in an IPv4 packet header.
IP precedence bits	The upper three bits in the: • Type of Service (ToS) byte of an IPv4 packet • Traffic Class byte of an IPv6 packet
IP precedence mode	See Type-of-Service (ToS), below.
IPv4	Version 4 of the IP protocol.
IPv6	Version 6 of the IP protocol.
outbound packet	A packet leaving the switch through any LAN port.
outbound port	Any port on the switch through which traffic leaves the switch.
outbound port queue	On any port, a buffer that holds outbound traffic until it is transmitted from the switch through the port. Depending on the switch, by default, there are either four or eight outbound queues for each port on the switch. Queue 4 or 8 is the highest priority queue; queue 1 is the lowest priority queue. Traffic in a port's high priority queue leaves the switch before any traffic in the port's medium or low priority queues.
QoS policy	Defines a policy configured in one of the following ways: • Classifier-based: Rate-limiting and/or prioritizing (by 802.1p priority, IP precedence, or DSCP bit setting) packets in a specified traffic class (defined by the **class** command). • Globally-configured: 802.1p priority and/or DSCP codepoint (with global QoS commands)
re-marking	Assigns a new QoS policy to an outbound packet by changing the: • Class-of-Service (CoS) 802.1p bit setting in Layer 2 VLAN headers • DSCP bit setting in the Layer 3 IPv4 ToS byte (or IPv6 Traffic Class byte).
tagged port membership	Identifies a port as belonging to a specific VLAN and enables VLAN-tagged packets to carry an 802.1p priority when sent from the port in outbound traffic. When a port is an untagged member of a VLAN, outbound packets belonging to the VLAN do not carry an 802.1p priority setting.

Term	Use in This Document
Type-of-Service (ToS)	One of the bytes in an IPv4 packet header. IPv4 packets may be classified according to two type-of-service modes: • IP-precedence mode, using the upper three bits of the ToS byte • Differentiated Services (Diff-Serv) mode, using the upper six bits of the ToS byte. For more information about the IPv4 ToS field, see Table 6-11 and the "IP-precedence bits" and "DSCP" definitions in this table.
Traffic Class byte	In IPv6 packets, the byte that corresponds to the IPv4 Type-of-Service field, consisting of a six-bit (high-order) Differentiated Services (Diff-Serv) field and a two-bit (low-order) reserved field. The three-bit (high-order) precedence field in the Traffic Class byte corresponds to the IP precedence bit set in IPv4 packets. See also "IP-precedence bits" and "DSCP" elsewhere in this table. ***Note***: As in IPv4, IPv6 packets may be classified according to two type-of-service modes: • IP-precedence mode, using the upper three bits of the Traffic Class byte • Differentiated Services (Diff-Serv) mode, using the upper six bits of the Traffic Class byte For more information about the IPv6 Traffic Class field, see Table 6-11 on page 6-89 and the "IP-precedence bits" and "DSCP" definitions in this table.
upstream device	A device linked directly or indirectly to an inbound switch port. The switch receives traffic from upstream devices.

QoS Operation

On the switches covered in this guide, QoS operation may be configured through a combination of the following methods:

- Globally-configured, switch-wide QoS settings
- Classifier-based per-port and per-VLAN QoS policies.

Classifier-based QoS policies are designed to work with existing globally-configured, switch-wide QoS settings by allowing you to zoom in on a subset of port or VLAN traffic to further manage it. You can use multiple match criteria to more finely select and define the classes of traffic that you want to manage. QoS policy actions determine how you can handle the selected traffic.

Note

While providing greater control for implementing QoS policies, classifier-based QoS policies may override globally-configured QoS settings. For more information, see "Override of Global QoS Settings" on page 6-81.

Be sure to carefully plan your QoS strategies in advance, identifying the network traffic that you can globally configure and the traffic on which you want to execute customized, classifier-based QoS actions.

Globally-Configured QoS

Globally-Configured QoS Feature	Default	Page Reference
UDP/TCP Priority	Disabled	page 6-24
IP-Device Priority	Disabled	page 6-33
IP Type-of-Service Priority	Disabled	page 6-41
LAN Protocol Priority	Disabled	page 6-54
VLAN-ID Priority	Disabled	page 6-56
Source-Port Priority	Disabled	page 6-62
DSCP Policy Table	Various	page 6-88
Queue Configuration	4 or 8 Queues*	page 6-95

*Series 3500yl switches use four queues by default.
Series 5400zl switches and the Series 8212zl switch use eight queues by default.

Globally-configured QoS operation supports the following types of packet classification and traffic marking on outbound port and VLAN traffic. For information on how to configure and use global QoS settings, see "Globally-Configured QoS" on page 6-19.

- Globally configured packet classification criteria include:
  - IPv4 device: source and destination address
  - Layer 2 802.1p priority (VLAN header)
  - Layer 3 protocol (such as ARP, IP, IPX, RIP)
  - Layer 3 IPv4 Type of Service (ToS) byte: IP precedence or DSCP bits
  - Layer 3 IPv6 Traffic Class byte: IP precedence or DSCP bits
  - Layer 4 UDP/TCP application port
  - Source port on the switch
  - VLAN ID
- Traffic marking options are as follows:
  - Setting the Layer 2 802.1p priority value in VLAN-tagged and untagged packet headers
  - Setting the Layer 3 Differentiated Services Codepoint (DSCP) bits in the ToS byte of IPv4 packet headers and Traffic Class byte of IPv6 headers.

Classifier-Based QoS

Classifier-Based QoS Feature	Default	Page Reference
Classifier-Based QoS Configuration Procedure	—	page 6-71
Configuring QoS Actions in a Policy	—	page 6-76
Override of Global QoS Settings	—	page 6-81
Viewing a Classifier-Based QoS Configuration	—	page 6-81
Classifier-Based QoS Restrictions	—	page 6-85
Classifier-Based QoS Configuration Examples		page 6-86
DSCP Policy Table	Various	page 6-88
Queue Configuration	4 or 8 Queues*	page 6-95

*Series 3500yl switches use four queues by default.
Series 5400zl switches and the Series 8212zl switch use eight queues by default.

Starting in release K.14.01, *classifier-based* QoS operation provides additional QoS actions on a per-port and per-VLAN basis.

- Classifier-based match criteria on inbound IPv4/IPv6 traffic include:
  - IP source address (IPv4 and IPv6)
  - IP destination address (IPv4 and IPv6)
  - IP protocol (such as ICMP or SNMP)
  - Layer 3 IP precedence bits
  - Layer 3 DSCP codepoint
  - Layer 4 UDP/TCP application port
  - VLAN ID
- Classifier-based QoS policy actions on matching IPv4/IPv6 packets are as follows:
  - Setting Layer 2 802.1p priority value (class of service) in VLAN-tagged and untagged packet headers
  - Setting the Layer 3 IP precedence bits
  - Setting the Layer 3 Differentiated-Services Codepoint (DSCP) bits
  - Rate-limiting inbound traffic on port and VLAN interfaces

For information on operation with globally-configured QoS settings, see “Advanced Classifier-Based QoS” on page 6-70.

QoS Packet Classification

To manage network traffic using QoS features, you must first classify (select) the packets you want to manage. You can use any combination of the following packet classification methods to select packets for QoS management:

- Globally configured, switch-wide classification criteria
- Classifier-based match criteria applied to inbound traffic on specific port and VLAN interfaces

IPv6 Support

Starting in software release K.14.01, global and classifier-based QoS policies support IPv6 packet classification in addition to IPv4.

Globally-Configured Packet Classification

Note On Using Multiple Global Criteria

ProCurve recommends that you configure a minimum number of global QoS classifiers to prioritize a specific packet type. Increasing the number of enabled global QoS classifiers increases the complexity of possible outcomes and consumes switch resources.

The switches covered in this guide provide six types of globally-configured QoS classifiers (match criteria) to select packets for QoS traffic marking.

When multiple, global QoS classifiers are configured, a switch uses the highest-to-lowest search order shown in Table 6-1 to identify the highest-precedence classifier to apply to any given packet. When a match between a packet and a classifier is found, the switch applies the QoS policy configured for the classifier and the packet is handled accordingly.

Table 6-1. Globally-Configured Packet Classification: Search Order and Precedence

Search Order	Precedence	Global QoS Classifier
1	1 (highest)	UDP/TCP application type (port)
2	2	Device priority (destination or source IP address)
3	3	IP type of service: precedence and DSCP bit sets (IP packets only)
4	4	IP protocol (IP, IPX, ARP, AppleTalk, SNA, and NetBeui)
5	5	VLAN ID

Search Order	Precedence	Global QoS Classifier
6	6	Incoming source-port on the switch
Default	7 (lowest)	The incoming 802.1p priority (present in tagged VLAN environments) is preserved if no global QoS classifier with a higher precedence matches.

Note that on the switches covered in this guide, if the switch is configured with multiple global classifiers that match the same packet, the switch only applies the QoS marking configured for the QoS classifier with the highest precedence. In this case, the QoS configuration for another, lower-precedence classifier that matches is ignored.

For example, if QoS assigns a high priority to packets belonging to VLAN 100 and normal priority to all IP protocol packets, because the IP protocol priority (4) has precedence over the VLAN priority (5), IP protocol packets on VLAN 100 are set to normal priority.

Classifier-Based Match Criteria

In classifier-based packet classification, *match criteria* provide a way to select the packets on which you want to execute QoS actions, such as rate-limiting or 802.1p prioritization.

Match criteria are configured by creating a *class* of IPv4 or IPv6 traffic, which contains one or more match or ignore statements. A traffic class may be used by any classifier-based software feature, such as QoS or port mirroring.

By using classifier-based QoS, you can configure multiple match criteria that search multiple fields in packet headers to select the exact traffic you want to rate-limit or prioritize for a port or VLAN interface. A classifier-based QoS policy is especially useful when you want to manage different types of traffic in the same way (for example, to prioritize both IP subnet and voice traffic).

For information on how to use match criteria to configure a traffic class, refer to “Classifier-Based Software Configuration” on page 9-1.

QoS Traffic Marking

As described in "QoS Operation" on page 6-9, when you apply or reconfigure QoS actions for selected packets, QoS supports different types of traffic marking in globally-configured QoS settings and classifier-based per-port or per-VLAN QoS policies.

Globally-Configured Traffic Marking

If a packet matches one of the globally-configured packet classifiers, QoS applies one of the following types of traffic marking to the outbound packet:

- **Layer 2 802.1p prioritization:** Controls the outbound port-queue priority for traffic leaving the switch, and (if traffic exits through a VLAN-tagged port) sends the priority setting with the individual packets to downstream devices.
- **Layer 3 DSCP marking:** Enables the switch to set, change, and honor prioritization policies by using the Differentiated Services (diff-serv) bits in the IPv4 ToS byte and IPv6 Traffic Class byte of packet headers.

Layer 2 802.1p Prioritization

By setting a new 802.1p priority value, QoS allows you to control the priority of outbound packets moving through the switch. The Layer 2 802.1p priority setting in a packet header determines the outbound port queue to which the packet is sent.

By default, a Series 3500yl switch has four queues; Series 5400zl switches and the Series 8212zl switch have eight outbound traffic queues (0 through 7). A lower-numbered queue has a lower outbound priority; a higher-numbered queue has a higher outbound priority. Packets are transmitted from the switch port on the basis of their queue assignment and whether any higher queues are empty. (To increase bandwidth, you can reconfigure the switch to use four or two outbound queues; see "QoS Queue Configuration" on page 6-95 for more information.)

Configuring a new 802.1p priority value allows you to set the outbound priority queue to which a packet is sent. For example, you can configure an 802.1p priority of 0 through 7 for an outbound packet. When the packet is sent to a port, the QoS priority determines the outbound queue to which the packet is assigned as shown in Table 6-2.

Table 6-2. 802.1p Priority Settings and Outbound Queue Assignment

802.1p Priority Setting	Outbound Port Queue
1 and 2	Low priority (1, 2)
0 or 3	Normal priority (3, 4)
4 and 5	Medium priority (5, 6)
6 and 7	High priority (7, 8)

If a packet is transmitted in an untagged-VLAN environment, the 802.1p priority settings in Table 6-2 control only the outbound queue to which the packet is sent on the local switch. Because no VLAN tag is used, an 802.1p priority value is not added to the 802.1Q field in the packet header for use by downstream devices.

However, if your network uses only one VLAN and does not require VLAN-tagged ports, you can preserve 802.1p priority settings in outbound traffic by configuring the ports on links between devices on which you want 802.1p priorities to be honored as tagged VLAN members.

If a packet is transmitted in an 802.1Q VLAN-tagged environment, the QoS-configured 802.1p setting is also added to the VLAN packet header as an 802.1p priority for use by downstream devices and applications (as shown in Table 6-3).

In an 802.1Q VLAN environment with VLAN-tagged ports, if QoS is *not* configured on the switch but *is* configured on an upstream device, the priorities carried in the packets determine the outbound port queue on which packets are forwarded.

Table 6-3. Mapping 802.1p Priorities to Outbound Port Queues on the Switch and Downstream Devices

Configured 802.1p Priority	Outbound Port Queue in the Switch	802.1p Priority Added to Tagged VLAN Packets Exiting the Switch	Queue Assignment in Downstream Devices With:		
			8 Queues	4 Queues	2 Queues
1	Queue 1	1 (low priority)	Queue 1	Queue 1	
2	Queue 2	2	Queue 2		Queue 1
0	Queue 3	0 (normal priority)	Queue 3	Queue 2	
3	Queue 4	3	Queue 4		
4	Queue 5	4 (medium priority)	Queue 5	Queue 3	
5	Queue 6	5	Queue 6		Queue 2
6	Queue 7	6 (high priority)	Queue 7	Queue 4	
7	Queue 8	7	Queue 8		

Note You can reconfigure the QoS queue setting to change the number of outbound port queues in the switch from eight or four (default) to four or two queues. For more information, see "QoS Queue Configuration" on page 6-95.

Layer 3 DSCP Marking

By changing or honoring the settings of the DSCP codepoint in IP packet headers, QoS allows you to control the DSCP and associated 802.1p priority values in outbound IP packets that are sent to downstream devices.

You can later configure downstream devices to read and use the DSCP policy that QoS sets. When marking the DSCP bits in IP packets, a QoS policy is not dependent on VLAN-tagged ports to carry 802.1p packet priorities to downstream devices (as shown in Table 6-4).

When you configure a Layer 3 DSCP policy, you specify:

- Bit values for the DSCP codepoint (the upper six bits in the ToS/Traffic Class byte in IP packet headers), entered in either binary format, the decimal equivalent, or an ASCII standard (hexadecimal) name
- An 802.1p priority value that is associated with the new DSCP bit values

  Certain DSCP codepoints (such as Assured Forwarding and Expedited Forwarding) have default 802.1p priorities as shown in Table 6-11.

A DSCP policy assigns a DSCP codepoint and 802.1p priority value to IPv4 and IPv6 packets. As shown in Figure 6-2, you can classify traffic on an edge switch and use Layer 3 DSCP-marking (instead of only 802.1p priority) to assign and preserve QoS policies on downstream devices. In this case, if you reconfigure the 802.1p priority associated with the DSCP codepoint, the new 802.1p assignment takes effect starting on the switch on which it is configured and is used in packets sent to downstream devices.

If you configure a different 802.1p priority for a DSCP codepoint, the new DSCP policy overrides the 802.1p priority value in packets which enter the switch with the specified codepoint. The Layer 2 802.1p priority setting (0 through 7) determines the outbound port queue to which a packet is sent (as shown in Table 6-2).

VLAN and Untagged VLAN Environments

QoS operates in VLAN-tagged and untagged environments. If your network does not use multiple VLANs, you can still implement the 802.1Q VLAN capability to allow packets to carry an 802.1p priority to the next downstream device. To do so, configure the ports on links to other network devices as VLAN-tagged members.

In a tagged or untagged VLAN, you can also ensure that IPv4/IPv6 packets carry an 802.1p priority to downstream devices by configuring DSCP marking in the ToS/Traffic Class byte.

Table 6-4 summarizes the QoS options for traffic-marking in VLAN-tagged and untagged environments.

Table 6-4. QoS Traffic Marking Supported in Tagged and Untagged VLANs

QoS Marking Supported on Outbound Packets	Port Membership in VLANs	
	Tagged	Untagged
Assign an 802.1p priority that determines the outbound port queue to which a packet is sent	Supported	Supported
Carry the 802.1p priority to the next downstream device	Supported	Not Supported
Carry a DSCP policy (DSCP codepoint[1] and associated 802.1p priority [2]) to downstream devices	Supported	Supported

[1] DSCP marking (DSCP codepoint and associated 802.1p priority) are not supported on non-IP packets and packets selected using the following global QOS classifiers: Layer 3 Protocol and IP-Precedence. Also, in order for DSCP policy marking to be honored on a downstream device, the device must be configured to use the DSCP policy in IP packet headers.

[2] The 802.1p priority associated with a DSCP codepoint (see Table 6-11) is used to determine the packet's outbound port queue. When used in a VLAN-tagged environment, an 802.1p priority is also carried in the 802.1Q field of outbound packet headers.

Classifier-Based Traffic Marking

Classifier-based per-port or per-VLAN QoS policies support the following traffic-marking actions. Note that in addition to globally-configured QoS traffic marking (802.1p and DSCP prioritization), classifier-based QoS policies also support IP precedence and rate-limiting.

- **Layer 2 802.1p prioritization:** Controls the outbound port queue priority for traffic leaving the switch, and (if traffic exits through a VLAN-tagged port) sends the priority setting in packet headers to downstream devices.
- **Layer 3 IP precedence-bit marking:** Enables the switch to set, change, and honor prioritization policies by using the IP precedence bits in the ToS byte of IPv4 packet headers and Traffic Class byte of IPv6 headers.
- **Layer 3 DSCP marking:** Enables the switch to set, change, and honor prioritization policies by using the Differentiated Services (diffserv) bits in the ToS byte of IPv4 headers and Traffic Class byte of IPv6 headers.
- **Rate-limiting:** Enables a port or VLAN interface to allow only the specified amount of bandwidth to be used for inbound traffic. When traffic exceeds the configured limit, it is dropped.

For information on how to configure and use classifier-based QoS policies, see "Advanced Classifier-Based QoS" on page 6-70.

Override of Global QoS Settings

After you apply a classifier-based QoS policy on a port or VLAN interface:

- The 802.1p (CoS) priority and DSCP codepoint marking applied to classified packets override any 802.1p and DSCP codepoint values that are globally-configured using the QoS commands, described in "Globally-Configured QoS" on page 6-19.
- The rate-limit applied to classified packets overrides any globally configured rate-limit globally-configured with the commands described in the "Port Traffic Controls" chapter in the *Management and Configuration Guide*.

For more information on how classifier-based traffic marking overrides globally-configured traffic marking, see "Override of Global QoS Settings" on page 6-81.

Globally-Configured QoS

QoS Feature	Default	Reference
UDP/TCP Priority	Disabled	page 6-24
IP-Device Priority	Disabled	page 6-33
IP Type-of-Service Priority	Disabled	page 6-41
Layer-3 Protocol Priority	Disabled	page 6-54
VLAN-ID Priority	Disabled	page 6-56
Source-Port Priority	Disabled	page 6-62

Note

For more information on how to use global QoS classifiers, see "Global QoS Restrictions" on page 6-22.

Global QoS Configuration Procedure

To globally configure a QoS policy on the switch, follow these steps:

1. Determine the global QoS policy you want to implement on the switch by analyzing the types of traffic flowing through your network and identifying one or more traffic types to prioritize. The order of precedence, from *a* (highest) to *h* (lowest), in which global QoS classifiers are applied is as follows:
   a. TCP/UDP applications
   b. Device priority—IP source or destination address (Note that destination has precedence over source. See Table 6-5.)
   c. IP precedence bit set (leftmost three bits in the ToS/Traffic Class field of IP packets)
   d. IP differentiated services bit set (leftmost six bits in the ToS/Traffic Class field of IP packets)
   e. Layer-3 protocol
   f. VLAN ID (requires at least one tagged VLAN on the network)
   g. Source port
   h. Incoming 802.1p priority (requires at least one tagged VLAN on the network)

Default: In a tagged VLAN environment, the incoming 802.1p priority is used as the default QoS classifier if no global QoS classifier with a higher precedence matches (see Table).

2. Select the global QoS classifier that you want to use. Table 6-5 shows the types of QoS marking (802.1p priority and/or DSCP codepoint) supported by each global QoS classifier.

Table 6-5. QoS Marking Supported by Global QoS Classifiers

Global QoS Classifiers	Type of QoS Marking Used to Prioritize Outbound Traffic	
	802.1p Priority[1] Only	DSCP Policy[2]: DSCP codepoint with 802.1p Priority
UDP/TCP	Supported	Supported
IP Device	Supported	Supported
IP Precedence	Supported[3]	Not Supported
IP DiffServ	Supported	Supported
L3 Protocol	Supported	Not Supported
VLAN ID	Supported	Supported
Source Port	Supported	Supported

[1] When you configure only the 802.1p priority to mark packets that match a global QoS classifier, the selected traffic is prioritized and sent to the corresponding outbound port queue on the switch (see Table 6-2). VLAN-tagged ports are necessary to carry the 802.1p priority in a packet header to downstream devices.

[2] When you configure a DSCP policy to mark packets that match a global QoS classifier, the selected traffic is also prioritized according to the associated 802.1p priority and sent to the corresponding outbound port queue on the switch. VLAN-tagged ports carry the 802.1p priority in a packet header to downstream devices. In addition, you can configure downstream devices to read the DSCP value in IP packets and implement the service policy implied by the codepoint.

[3] When using a global QoS IP Precedence classifier, the 802.1p priority is automatically assigned to matching packets based on the IP precedence bit set in the packet header.

3. If you want 802.1p priority settings to be included in outbound packets, ensure that tagged VLANs are configured on the appropriate downstream links.

4. Determine the global QoS policy required for each QoS-capable device in your network and configure the necessary settings.

   If you want downstream devices to recognize and use DSCP codepoints in IP packets sent from the switch, enable ToS Differentiated Service mode on the devices and configure the appropriate DSCP policies. Note that certain DSCP policies have a default 802.1p priority automatically assigned (see Table 6-13).

Viewing a Global QoS Configuration

To display the existing switch-wide configurations for a global QoS classifier, use one of the following **show qos** commands.

Syntax: show qos < *global-classifier* >

tcp-udp-port-priority

Displays the current TCP/UDP port priority configuration. Refer to figure 6-7 on page 6-31.

device-priority

Displays the current device (IP address) priority configuration. Refer to figure 6-9 on page 6-35.

type-of-service

Displays the current type-of-service priority configuration. The display output differs according to the option used:

- *IP Precedence: Refer to figure 6-14 on page 6-44.*
- *Diffserve: Refer to figure 6-16 on page 6-48.*

protocol-priority

Displays the current protocol priority configuration.

vlan-priority

Displays the current VLAN priority configuration. Refer to figure 6-23 on page 6-57.

port-priority

Displays the current source-port priority configuration. Refer to figure 6-28 on page 6-63.

No Override

By default, the **show qos** output for following global QoS classifiers may display **No-override** for QoS marking: IP Precedence, IP Diffserv, Layer-3 Protocol, VLAN ID, and Source-port (see Figure 6-3). **No-override** means that the global QoS policy used to mark matching packets does not assign an 802.1p value.

- IP packets received through a VLAN-tagged port are managed using the 802.1p priority they carry in the 802.1Q field in their headers.
- VLAN-tagged packets received through an untagged port are handled by the switch with "normal" priority.

For example, Figure 6-3 below shows the global QoS configurations on the switch that are configured with the VLAN ID classifier. Note that non-default 802.1p priorities have been configured for VLAN IDs 22 and 33; packets received on VLAN 1 are managed with the default settings, as described in the two bulleted items above.

```
ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       No-override |        No-override
  22      Priority    |        0
  33      DSCP        | 000010 6
```

In the **show** output, VLAN 1 uses the default priority values; VLANs 22 and 33 are configured to mark packets with new 802.1p and DSCP values.

Figure 6-3. Example of the Show QoS Command Output

Global QoS Restrictions

Table 6-3 shows the packet types supported by different global QoS classifiers and DSCP marking.

Table 6-6. Restrictions for Global QoS Support

Type of Packets Supported	Global QoS Classifiers							DSCP Overwrite (Re-Marking)
	TCP/UDP	IP Device	IP Type-of-Service	Layer 3 Protocol	VLAN ID	Source Port	Incoming 802.1p	
IP packets (IPv4 and IPv6[1]) only	Yes	Yes	Yes	No	No	No	No	Yes
Layer-2 SAP encapsulation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[1]Globally-configured QoS supports IPv6 packets starting in release K.14.01.

- **All Switches:** For explicit QoS support of IP subnets, ProCurve recommends forcing IP subnets onto separate VLANs and then configuring VLAN-based classifiers for those VLANs.
- **For Devices that Do Not Support 802.1Q VLAN-Tagged Ports:** For communication between these devices and the switch, connect the device to a switch port configured as **Untagged** for the VLAN in which you want the device's traffic to move.
- **Port Tagging Rules:** For a port on the switch to be a member of a VLAN, the port must be configured as either **Tagged** or **Untagged** for that VLAN. A port can be an untagged member of only one VLAN of a given protocol type. Otherwise, the switch cannot determine which VLAN should receive untagged traffic. For more on VLANs, refer to chapter 2, "Static Virtual LANs (VLANs)".
- **Maximum Global QoS Remarking Entries:** The switches covered in this guide accept the maximum number of configured outbound 802.1p priority and DSCP entries shown in Table 6-7.

Table 6-7. Maximum Number of QoS Entries.

Switch	Software Version	Maximum QoS Remarking	Notes
Switch 8212zl Series 5400zl			• Each IP Device (IP address) QoS configuration uses two entries. • Each TCP/UDP Port QoS configuration uses two entries. • All other global QoS classifier configurations use one entry each.
Series 5300yl		250* configured entries	
*Configuring IP Device (IP address) and TCP/UDP global QoS classifiers reduces this maximum. For more information, see the "Notes" column.			

If the global QoS configurations on a switch exceed the maximum number of entries shown in Table 6-7, the following error message is displayed:

```
Unable to add this QoS rule. Maximum number (entry-#)
already reached.
```

- **Not Supported:** Use of an inbound 802.1p packet priority as a classifier for remapping a packet's outbound priority to different 802.1p priority. For example, where inbound packets carry an 802.1p priority of 1, QoS cannot be configured use this priority as a classifier for changing the outbound priority to 0.

- **Fragmented Packets & TCP/UDP:** QoS is not performed on fragmented packets under TCP/UDP
- **Monitoring Shared Resources:** The QoS feature shares internal switch resources with several other features. The switch provides ample resources for all features. However, if the internal resources become fully subscribed, additional QoS provisions cannot be configured until the necessary resources are released from other uses. For information on determining the current resource availability and usage, refer to the appendix titled "Monitoring Resources" in the *Management and Configuration Guide* for your switch.

Global TCP/UDP Classifier

Global QoS Classifier Precedence: 1

When you use TCP or UDP and a Layer 4 Application port number as a global QoS classifier, traffic carrying the specified TCP/UDP port number(s) is marked with a specified priority level, without regard for any other QoS classifiers in the switch. You can configure up to 50 TCP/UDP application port numbers as QoS classifiers.

Note

Starting in software release K.14.01, global TCP/UDP classifiers are supported on IPv4, IPv6, or both IPv4 and IPv6 packets. In previous releases, only IPv4 packets were supported.

Options for Assigning Priority. The packet-marking options for global TCP/UDP port-number classifiers include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets can be IPv4 or IPv6.)

For a given TCP or UDP port number, you can use only one of the above options at a time. However, for different port numbers, you can use different options.

TCP/UDP Port Number Ranges. There are three ranges:

- Well-Known Ports: 0 - 1023
- Registered Ports: 1024 - 49151
- Dynamic and/or Private Ports: 49152 - 65535

For more information, including a listing of UDP/TCP port numbers, go to the *Internet Assigned Numbers Authority* (IANA) website at:

www.iana.org

Then click on:

Protocol Number Assignment Services

P (Under "Directory of General Assigned Numbers" heading)

Port Numbers

Assigning an 802.1p Priority for a Global TCP/UDP Classifier

To mark matching TCP or UDP packets with an 802.1p priority, enter the following command:

Syntax: qos < udp-port | tcp-port > [ipv4 | ipv6 | ip-all] <*port-number* | range *start end* > priority < 0 - 7 >

Marks an 802.1p priority in outbound packets with the specified TCP or UDP application-port number, where:

- **ipv4** *marks only IPv4 packets (default).*
- **ipv6** *marks only IPv6 packets.*
- **ip-all** *marks all IP traffic (both IPv4 and IPv6 packets).*
- ***port-number*** *is a TCP/UDP port number from 1 to 65535.*
- **range *start end*** *specifies a range of TCP/UDP ports; see "Operating Notes on Using TCP/UDP Port Ranges" on page 6-26. If you specify a range, the minimum port number must precede the maximum port number in the range.*
- **priority <0-7>** *marks the specified 802.1p priority in matching TCP or UDP packets.*

The 802.1p priority determines the packet's queue in the outbound port on the switch. If the packet leaves the switch on a tagged VLAN port, it carries the 802.1p priority with it to the next downstream device.

Default: Disabled — No 802.1p priority is assigned.

The **no** *form of the command deletes the specified UDP or TCP port number or range of port numbers as a QoS classifier.*

Note: *If you have specified a range of port numbers, you must specify the entire range in the* **no** *command; you cannot remove part of a range.*

show qos tcp-udp-port-priority

Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.

Operating Notes on Using TCP/UDP Port Ranges

- You can only have 6 concurrent policies when using unique ranges. The number of policies allowed is lower if ACLs are also using port ranges.
- You cannot have ranges that include any port numbers that have been configured as part of another QoS application port number policy.
- An error message is generated if there are not enough hardware resources available when configuring a policy.
- You must specify the entire range of configured port numbers when using the **no** form of the command, for example:

```
ProCurve(config)# qos udp-port range 1300 1399 dscp 001110
ProCurve(config)# no qos range 1300 1399
```

Example. The following example displays the following configuration for TCP and UDP port prioritization:

TCP/UDP Port	802.1p Priority for TCP	802.1p Priority for UDP
TCP Port 23 (Telnet)	7	7
UDP Port 23 (Telnet)	7	7
TCP Port 80 (World Wide Web HTTP)	2	2
UDP Port 80 (World Wide Web HTTP)	1	1

```
ProCurve(config)# qos tcp-port 23 priority 7
ProCurve(config)# qos tcp-port 80 priority 2
ProCurve(config)# qos udp-port 23 priority 7
ProCurve(config)# qos udp-port 80 priority 1
ProCurve(config)# qos udp-port range 100 199 priority 3
ProCurve(config)# show qos tcp-udp-port-priority

  TCP/UDP port based priorities

           | IP Packet Application              |
  Protocol | Type      Port        Apply rule | DSCP   Priority
 -------- + --------- ----------- ---------- + ------ -----------
  TCP      | IPV4      23          Priority   |        7
  TCP      | IPV4      80          Priority   |        2
  UDP      | IPV4      23          Priority   |        7
  UDP      | IPV4      80          Priority   |        1
  UDP      | IPV4      100-199     Priority   |        3
```

Values in these two columns define the QoS classifiers used to select the packets to prioritize.

Indicates that 802.1p priority assignments are in use for packets with 23, 80 or 100-199 as a TCP or UDP port number.

Displays the 802.1p priority assignment for packets with the indicated QoS classifiers.

Figure 6-4. Configuring 802.1p Priority Assignments on TCP/UDP Ports

Assigning a DSCP Policy for a Global TCP/UDP Classifier

This global QoS packet-marking option assigns a previously configured or default DSCP policy (codepoint and 802.1p priority) to TCP or UDP packets having the specified port number or range of port numbers. When assigning a DSCP policy, the switch performs the following actions:

1. Selects an incoming IP packet if the TCP or UDP port number it carries matches the port number specified in the TCP or UDP classifier (as shown in figure 6-4, above).
2. Overwrites (re-marks) the packet's DSCP with the new DSCP configured for matching packets.
3. Assigns the 802.1p priority associated with the new DSCP. (See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88.)
4. Forwards the packet through the appropriate outbound port queue.

Creating a DSCP Policy Based on TCP/UDP Port Number Classifiers.

The following procedure creates a DSCP policy for IP packets carrying the selected TCP or UDP port-number classifier.

1. Identify the TCP or UDP port-number classifier you want to use for assigning a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected TCP or UDP port number or range of port numbers.
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite (re-mark) the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.

3. If necessary, use the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.

Prerequisite

A DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows **No-override** in the Priority column of the DSCP Policy table (**show qos dscp-map** command), you must first configure a priority for the codepoint before proceeding (**qos dscp-map priority** command). See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88 for more information.

Syntax: qos dscp-map < *codepoint* > priority <0 - 7>>

*(Optional) This command is required only if an 802.1p priority is not already assigned to the specified <**codepoint**> in the DSCP Policy table (see Table 6-11 on page 6-89).*

Valid values for a DSCP codepoint are as follows:
- A binary value for the six-bit codepoint from **000000** *to* **111111**.
- A decimal value from **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
-An ASCII standard name for a binary DSCP bit set:

af11 (001010)	**af42** (100100)
af12 (001100)	**af43** (100110)
af13 (001110)	**ef** (101110)
af21 (010010)	**cs1** (001000) = *precedence 1*
af22 (010100)	**cs2** (010000) = *precedence 2*
af23 (010110)	**cs3** (011000) = *precedence 3*
af31 (011010)	**cs4** (100000) = *precedence 4*
af32 (011100)	**cs5** (101000) = *precedence 5*
af33 (011110)	**cs6** (110000) = *precedence 6*
af41 (100010)	**cs7** (111000) = *precedence 7*
default (000000)	

Type **?** *to display the list of valid codepoint entries.*

When the switch applies the specified DSCP policy to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IP packets, the DSCP will be replaced by the codepoint specified in this command. (Default: **No-override** *for most codepoints. See Table 6-11.)*

4. Configure the switch to assign the DSCP policy to packets with the specified TCP or UDP port number or range of port numbers.

Syntax: [no] qos <udp-port | tcp-port> [ipv4 | ipv6 | ip-all] <*port-number* | range *start end* > <dscp <*codepoint* >

Assigns a DSCP policy to outbound packets having the specified TCP or UDP application-port number or port range, and overwrites the DSCP in these packets with the assigned **<*codepoint*>** *value, where:*

- **ipv4** *marks only IPv4 packets (default).*
- **ipv6** *marks only IPv6 packets.*
- **ip-all** *marks all IP traffic (both IPv4 and IPv6 packets).*
- ***port-number*** *specifies a TCP/UDP port-number from 1 to 65535.*
- **range** ***start end*** *specifies a range of TCP/UDP ports; see "Operating Notes on Using TCP/UDP Port Ranges" on page 6-26. If you specify a range, the minimum port number must precede the maximum port number in the range.*
- **dscp** ***codepoint*** *overwrites the DSCP codepoint in the IPv4 ToS byte or IPv6 Traffic Class byte of matching packets with the specified value.*

  Valid values for the DSCP codepoint are as follows:
  - A binary value for the six-bit codepoint from **000000** *to* **111111**.
  - A decimal value from **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
  -An ASCII standard name for a binary DSCP bit set
  Type **?** *to display the list of valid codepoint entries.*

  The DSCP value you enter must be currently associated with an 802.1p priority in the DSCP Policy table (see Table 6-10). The 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device.

  The default DSCP codepoint is **No-override**. *The DSCP codepoint is not overwritten in matching packets.*

The **no** *form of the command deletes the specified UDP or TCP port number or range of port numbers as a QoS classifier. If you configured a range of port numbers as the QoS classifier, you must enter the entire range in the* **no** *command; you cannot remove part of a range.*

Syntax: show qos tcp-udp-port-priority

Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.

Example. This example shows how to assign the following DSCP policies to packets that match the specified TCP and UDP port applications:

Port Applications	DSCP Policies	
	DSCP	Priority
23-UDP	000111	7
80-TCP	000101	5
914-TCP	000010	1
1001-UDP	000010	1

1. Determine if the DSCP codepoints that you want to use to mark matching packets already have an 802.1p priority assigned, which could indicate use by existing applications (**show qos dscp-map** command).

   Note that a DSCP codepoint must also have a priority configured before you can use it to mark matching packets.

```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .           .
     .           .
     .           .
```

The DSCP codepoints have not yet been assigned an 802.1p priority level.

Figure 6-5. Displaying the Current DSCP-Map Configuration

2. Configure the DSCP policies for the codepoints you want to use.

```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
```

New DSCP policies have been configured

Figure 6-6. Assigning Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected TCP/UDP port applications and display the result.

```
ProCurve(config)# qos udp-port 23 dscp 000111
ProCurve(config)# qos tcp-port 80 dscp 000101
ProCurve(config)# qos tcp-port 914 dscp 000010
ProCurve(config)# qos udp-port range 1001 2000 dscp 000010

  TCP/UDP port based priorities

            | IP Packet Application              |
  Protocol  | Type      Port        Apply rule   | DSCP   Priority
  --------- + --------- ----------- ------------ + ------ -----------
  UDP       | IPV4      23          DSCP         | 8      7
  TCP       | IPV4      80          DSCP         | 6      5
  TCP       | IPV4      914         DSCP         | 3      1
  UDP       | IPV4      1001-2000   DSCP         | 3      1
```

Global TCP/UDP port-number classifiers

DSCP Policy: DSCP codepoint (3) and 802.1p priority (1) mapping (Note: DSCP 3 is the decimal equivalent of binary 000010.)

Figure 6-7. Configuring a DSCP Policy for Global TCP/UDP Port Classifiers

The switch applies the DSCP policies in Figure 6-7 to IP packets with the specified TCP/UDP port applications that are received in the switch. The switch manages the packets as follows:

- Overwrites the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assigns the 802.1p priorities in the above policies to the selected packets.

Displaying Resource Usage for QoS Policies

When you configure global QoS classifiers using TCP/UDP and a Layer 4 Application port number or port range, the switch automatically assigns two QoS resources for each policy—one for traffic to the TCP/UDP destination port and one for traffic to the TCP/UDP source port.

The **show qos resources** command displays the number of hardware resources currently in use by QoS policies as well as other software features.

Incldues the hardware resources used by currently applied QoS policies.

```
ProCurve# show qos resources

 Resource usage in Policy Enforcement Engine
        |    Rules    |  Rules Used
  Ports |  Available  |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+-------------+-------+-------+-------+-------+--------+-------|
  1-24  |        3014 |    15 |    11 |     0 |     1 |      0 |     3 |
  25-48 |        3005 |    15 |    10 |    10 |     1 |      0 |     3 |
  A     |        3017 |    15 |     8 |     0 |     1 |      0 |     3 |

        |    Meters   |  Meters Used
  Ports |  Available  |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+-------------+-------+-------+-------+-------+--------+-------|
  1-24  |         250 |       |     5 |     0 |       |        |     0 |
  25-48 |         251 |       |     4 |     0 |       |        |     0 |
  A     |         253 |       |     2 |     0 |       |        |     0 |

        | Application |
        | Port Ranges |  Application Port Ranges Used
  Ports |  Available  |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+-------------+-------+-------+-------+-------+--------+-------|
  1-24  |        3014 |     2 |     0 |     0 |       |      0 |     0 |
  25-48 |        3005 |     2 |     0 |     0 |       |      0 |     0 |
  A     |        3017 |     2 |     0 |     0 |       |      0 |     0 |

 0 of 8 Policy Engine management resources used.
 Key:
 ACL = Access Control Lists
 QoS = Device & Application Port Priority, QoS Policies, ICMP rate limits
 IDM = Identity Driven Management
 VT  = Virus Throttling blocks
 Mirror = Mirror Policies, Remote Intelligent Mirror endpoints
 Other = Management VLAN, DHCP Snooping, ARP Protection, Jumbo IP-MTU.

 Resource usage includes resources actually in use, or reserved for future
 use by the listed feature.  Internal dedicated-purpose resources, such as
 port bandwidth limits or VLAN QoS priority, are not included.
```

Figure 6-8. Displaying the Hardware Resources Used by Currently Configured QoS Policies

Note

ACLs and QoS policies share the same application port ranges. If a new QoS policy specifies a port range that is already configured for one or more ACLs, the QoS column increases by 1, but the "Application Port Ranges Available" column remains unchanged. Likewise, if an ACL is configured for a port range on which a QoS policy is already applied, the ACL column increases by 1, while the "Available" column remains unchanged.

Similarly, when you remove a port range, the "Application Port Ranges Available" column only increases if the port range is not configured for an existing ACL or QoS policy on the switch.

Global IP-Device Classifier

Global QoS Classifier Precedence: 2

The global IP-device classifier enables you to configure up to 300 IP addresses to select IP packets according to source *or* destination address.

IPv6 Support

Starting in software release K.14.01, IP device classifiers are supported on IPv4, IPv6, and IPv4/IPv6 subnets. In previous releases, only IPv4 packets are supported.

When a globally-configured IP-device address has the highest precedence in the switch for traffic addressed to or from the device, traffic received on the switch with the configured IP address is marked with the specified priority level. You can configure different IP-device classifiers with different priority levels.

QoS IP-Device Restriction

The configuration of a QoS IP-device priority on the Management VLAN IP address (if configured) is not supported. If no Management VLAN is configured, the configuration of a QoS IP-device priority on the default VLAN IP address is not supported.

Options for Assigning Priority. The packet-marking options for global IP-device classifiers include:

- 802.1p priority
- DSCP policy: Assigning a new DSCP and 802.1p priority

For information on global QoS operation when other global classifiers apply to the same traffic, see to "Globally-Configured Packet Classification" on page 6-12.

For a given IP address or subnet mask, you can assign only one of the above options at a time. However, for different IP addresses, you can use different options.

Assigning a Priority for a Global IP-Device Classifier

This global QoS packet-marking option assigns an 802.1p priority to all IP packets that have the specified IP address as either a source or destination. If both the source and destination addresses match, the priority configured for the IP destination address has precedence.

IPv4 Syntax: qos device-priority < *ipv4-address* | [ipv4] *ipv4-address/mask-length* > priority < 0 - 7 >

IPv6 Syntax: qos device-priority < *ipv6-address* | ipv6 *ipv6-address/prefix-length* >> priority < 0 - 7 >

Marks an 802.1p priority in outbound packets with the specified IP address or subnet mask in the source or destination field in a packet header, where:

- ***ipv4-address*** *or* ***ipv6-address*** *is an IPv4 or IPv6 address used to match the source or destination address in packet headers.*

  Note: *An IPv6 local-link address (such as fe80::110:252%vlan20) that is automatically generated on a VLAN interface is not supported as an* ***ipv6-address*** *value.*
- **[ipv4]** ***ipv4-address/mask-length*** *is the subnet identified by the IPv4 mask for the specified address that is used to match the IPv4 in the source or destination field of packet headers.*
- **ipv6** ***ipv6-address/prefix-length*** *is the subnet identified by the IPv6 prefix-length for the specified address that is used to match the IPv6 address in the source or destination field of packet headers.*

  Enter the IPv4 mask or IPv6 prefix length with an address in CIDR format by using the number of significant bits (for example, 2001:db8::1:262:a03:e102:127/64 or 10.28.31.1/24).
- **priority <0-7>** *marks the specified 802.1p priority in matching IP packets.*

The 802.1p priority determines the packet's queue in the outbound port on the switch. If the packet leaves the switch on a tagged VLAN port, it carries the 802.1p priority with it to the next downstream device.

The **no** *form of the command deletes the specified IP address or subnet mask as a QoS classifier, and resets the priority for the VLAN to* **No-override**.

show qos device-priority

Displays a listing of all IP device-priority QoS configurations currently in the running-config file.

Example. This example shows how to configure and display the 802.1p priority used to mark packets that match each global IP-device classifier:

IP Address / Mask or Prefix Length	802.1p Priority
10.28.31.1	7
10.28.31.130	5
10.28.31.100/24	1
2001:db8:2:1:212:79ff:fe88:a100	3
2001:db8:3:3::/64	1

```
ProCurve(config)# qos device-priority 10.28.31.1 priority 7
ProCurve(config)# qos device-priority 10.28.31.130 priority 5
ProCurve(config)# qos device-priority ipv4 10.28.32.100/24 priority 1
ProCurve(config)# qos device-priority 2001:db8:2:1:212:79ff:fe88:a100 priority 3
ProCurve(config)# qos device-priority ipv6 2001:db8:3:3::/64 priority 1
ProCurve(config)# show qos device-priority

  Device priorities

  Device Address                                Apply rule | DSCP   Priority
  --------------------------------------------  ---------- + ------ -----------
  10.28.31.1                                    Priority   |        7
  10.28.31.130                                  Priority   |        5
  10.28.32.100/24                               Priority   |        1
  2001:db8:2:1:212:79ff:fe88:a100               Priority   |        3
  2001:db8:3:3::/64                             Priority   |        1
```

Figure 6-9. Using a Global IP-Device Classifier to Mark Matching Packets

Assigning a DSCP Policy For a Global IP-Device Classifier

This global QoS packet-marking option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified IP address or subnet mask in the source or destination field of their packet header. The switch:

1. Selects an incoming IPv4 or IPv6 packet on the basis of the source or destination IP address or subnet mask it carries.
2. Overwrites the DSCP in matching packets with the globally configured DSCP codepoint, and assigns the 802.1p priority associated with the new DSCP. (See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88.)
3. Forwards the packet through the appropriate outbound port queue.

For more on DSCP, refer to "QoS Terminology" on page 6-7.

Creating a Policy Based on IP Address. This procedure creates a DSCP policy for IP packets carrying the selected IP address (source or destination).

1. Identify the IPv4 or IPv6 address to use as a classifier for assigning a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected IP address:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.
3. If necessary, use the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.

Prerequisite

A DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows **No-override** in the Priority column of the DSCP Policy table (**show qos dscp-map** command), you must first configure a priority for the codepoint before proceeding (**qos dscp-map priority** command). See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88 for more information.

Syntax: qos dscp-map < *codepoint* > priority <0 - 7>>

*(Optional) This command is required only if an 802.1p priority is not already assigned to the specified <**codepoint**> in the DSCP Policy table (see Table 6-11 on page 6-89).*

When the switch applies this policy to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IP packets, the DSCP will be replaced by the codepoint specified in this command. (Default: **No-override** *for most codepoints. See Table 6-11.)*

4. Configure the switch to assign the DSCP policy to packets with the specified IP address or subnet mask.

IPv4 Syntax: qos device-priority < *ipv4-address* | [ipv4] *ipv4-address/mask-length* > dscp < *codepoint* >

IPv6 Syntax: qos device-priority < *ipv6-address* | ipv6 *ipv6-address/prefix-length* >> dscp < *codepoint* >

Assigns a DSCP policy in packets with the specified IP address or subnet mask in the source or destination field in a packet header, where:

- ***ipv4-address*** *or* ***ipv6-address*** *is an IPv4 or IPv6 address used to match the source or destination address in packet headers.*

  Note: *An IPv6 local-link address (such as fe80::110:252%vlan20) that is automatically generated on a VLAN interface is not supported as an* ***ipv6-address*** *value.*
- **[ipv4]** ***ipv4-address/mask-length*** *is the subnet identified by the IPv4 mask for the specified address that is used to match the IPv4 in the source or destination field of packet headers.*
- **ipv6** ***ipv6-address/prefix-length*** *is the subnet identified by the IPv6 prefix-length for the specified address that is used to match the IPv6 address in the source or destination field of packet headers.*

  Enter the IPv4 mask or IPv6 prefix length with an address in CIDR format by using the number of significant bits (for example, 2001:db8:2:1:262:a03:e102:127/64 or 10.28.31.1/24).

- **dscp *codepoint*** *overwrites the DSCP codepoint in the IPv4 ToS byte or IPv6 Traffic Class byte of matching packets with the specified value.*

  Valid values for the DSCP codepoint are as follows:
  - A binary value for the six-bit codepoint from **000000** *to* **111111**.
  - A decimal value from **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
  -An ASCII standard name for a binary DSCP bit set
  Type **?** *to display the list of valid codepoint entries.*

  The DSCP value you enter must be currently associated with an 802.1p priority in the DSCP Policy table (see Table 6-10). The 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device.

  The default DSCP codepoint is **No-override**. *The DSCP codepoint is not overwritten in matching packets.*

The **no** *form of the command deletes the specified IP address or subnet mask as a QoS classifier. If you configured a subnet mask as match criteria, you must enter the entire IP address and mask-length in the* **no** *command.*

show qos device-priority

Displays a listing of all IP addresses and subnet masks used as QoS classifiers currently in the running-config file.

Example. This example shows how to assign the following DSCP policies to the packets that match the specified global IP-device classifiers:

IP Address	DSCP Policy	
	DSCP Codepoint	802.1p Priority
10.28.31.1	000111	7
10.28.31.130	000101	5
10.28.31.100/24	000010	1
2001:db8:2:1:212:79ff:fe88:a100	000101	3
2001:db8:3:3::/64	000010	1

1. Determine if the DSCP codepoints that you want to use to mark matching packets already have an 802.1p priority assigned, which could indicate use by existing applications (**show qos dscp-map** command). This is not a

problem if the configured priorities are acceptable for all applications that use the same DSCP (see "Notes on Changing a Priority Setting" on page 6-91).

Note that a DSCP codepoint must have an associated priority before you can use it to mark matching packets.

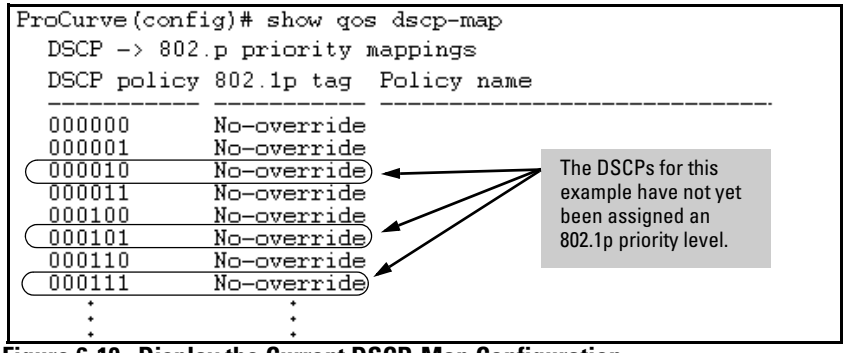


Figure 6-10. Display the Current DSCP-Map Configuration

2. Configure the priorities for the DSCPs you want to use to mark packets.

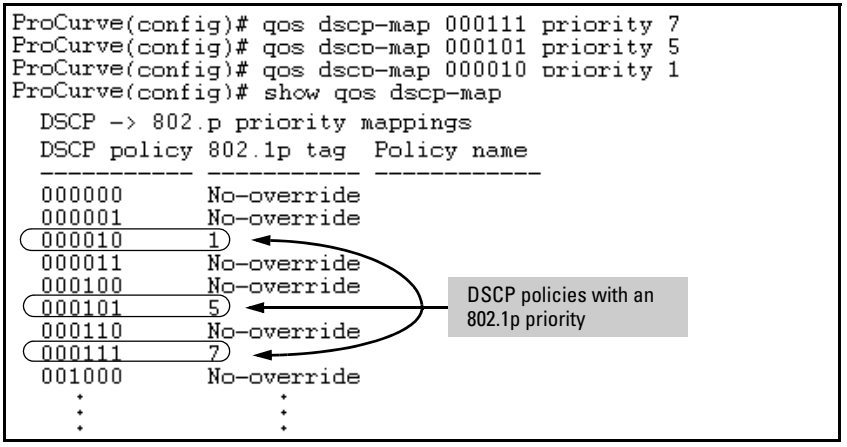


Figure 6-11. Assigning 802.1p Priorities to the Selected DSCPs

3. Assign the DSCP policies to the specified IP-device addresses and display the result.

```
ProCurve(config)# qos device-priority 10.28.31.1 dscp 000111
ProCurve(config)# qos device-priority 10.28.31.130 dscp 000101
ProCurve(config)# qos device-priority ipv4 10.28.32.100/24 dscp 000010
ProCurve(config)# qos device-priority 2001:db8:2:1:212:79ff:fe88:a100 dscp 0001
ProCurve(config)# qos device-priority ipv6 2001:db8:3:3/64 dscp 000010
ProCurve(config)# show qos device-priority

  Device priorities

  Device Address                                 Apply rule | DSCP   Priority
  ---------------------------------------------  ---------- + ------ ----------
  10.28.31.1                                     Priority   | 000111 7
  10.28.31.130                                   Priority   | 000101 5
  10.28.32.100/24                                Priority   | 000010 1
  2001:db8:2:1:212:79ff:fe88:a100                Priority   | 000101 3
  2001:db8:3:3/64                                Priority   | 000010 1
```

Figure 6-12. The Completed Device-Priority/Codepoint Configuration

The switch applies the DSCP policies in Figure 6-11 to IP packets with the specified IP addresses and subnet masks (source or destination) received in the switch. The switch manages the packets as follows:

- Overwrites the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assigns the 802.1p priorities in the above policies to the appropriate packets.

Global IP Type-of-Service Classifier

Global QoS Classifier Precedence: 3

The global IP Type-of-Service classifier enables you to classify and mark IP packets according to the following modes:

- **IP-Precedence Mode:** All IP packets generated by upstream devices and applications include a precedence bit set in the ToS/Traffic Class byte. In IP-precedence mode, the switch uses the precedence bits to compute and assign the corresponding 802.1p priority.
- **IP Differentiated Services (Diffserv) Mode:** The Diffserv mode uses the codepoints set in IP packets by upstream devices and applications to assign an 802.1p priority to packets. You can use Diffserv mode to mark packets in the following ways:
  - **Assign a new DSCP policy:** A "policy" includes both a codepoint and a corresponding 802.1p priority. This option selects an incoming IP packet on the basis of its codepoint and assigns a new codepoint and corresponding 802.1p priority. (Use the **qos dscp-map** command to specify a priority for any codepoint—page 6-88.)
  - **Assign an 802.1p priority:** This option reads the DSCP of an incoming IP packet and, without changing this codepoint, assigns the 802.1p priority to the packet, as configured in the DSCP Policy Table (page 6-88). This means that a priority value of 0 - 7 must be configured for a DSCP before the switch can perform a QoS match on the packet's DSCP bits.

Prerequisite

A DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows **No-override** in the Priority column of the DSCP Policy table (**show qos dscp-map** command), you must first configure a priority for the codepoint before proceeding (**qos dscp-map priority** command). See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88 for more information. Note that some 802.1p priorities are assigned by default to well-known DSCP codepoints, such as the "Assured Forwarding" and "Expedited Forwarding" codepoints (see Table 6-11).

Unless IP-Precedence mode and Diffserv mode are both disabled (the default setting), enabling one automatically disables the other. For more information on Type-of-Service operation, refer to "IPv4 ToS/IPv6 Traffic Class Byte" on page 6-42.

IPv4 ToS/IPv6 Traffic Class Byte

IPv4 packet headers contain a Type of Service (ToS) byte; IPv6 packet headers contain a Traffic Class byte. In an IPv6 packet, the Traffic Class byte is used in the same way as the ToS byte in an IPv4 packet. A ToS/Traffic Class byte includes a DSCP codepoint and precedence bits:

- **Differentiated Services Codepoint (DSCP):** Consists of the upper six bits of the ToS/Traffic Class byte. There are 64 possible codepoints.

  In the switches covered in this guide, the default QoS configuration includes some codepoints, such as Assured Forwarding and Expedited Forwarding, that are preconfigured with an 802.1p priority setting. All other codepoints are not configured with an 802.1p priority and display **No-override** as shown in the default DSCP Policy table (Table 6-11).

  Use the **qos dscp map** command to configure the switch to assign different 802.1p priorities to IP packets with different codepoints. Also, you can configure the switch to assign a new codepoint with its associated priority level (0-7) to matching packets as follows:

  a. Configure a DSCP codepoint with the desired priority in an edge switch.

  b. Configure the local switch to mark specified inbound traffic with the DSCP (and thus create a policy for that traffic type).

  c. Configure the internal switches in your LAN to honor the policy.

  For example, you could configure an edge switch to assign a codepoint of 000001 to all packets received from a specific VLAN, and then handle all traffic with that codepoint at high priority.

  For a codepoint listing and the commands for displaying and changing the DSCP Policy table, refer to “Differentiated Services Codepoint (DSCP) Mapping” on page 6-88.

- **Precedence Bits:** A subset of the DSCP codepoint, consisting of the upper three bits of the ToS/Traffic Class byte. When a global IP-Precedence classifier is configured, the switch uses the precedence bit set to determine the priority for selected packets as shown in Table 6-8. (The switch does not change the setting of the precedence bits.)

Table 6-8. IP Precedence-to-802.1p Priority Mapping

ToS/Traffic Class Byte: IP Precedence Bits	Corresponding 802.1p Priority	Service Priority Level
000	1	Lowest
001	2	Low
002	0	Normal
003	3	
004	4	
005	5	
006	6	
007	7	Highest

Note

Using a global IP-Precedence classifier to prioritize IP packets relies on priorities set in upstream devices and applications.

Figure 6-13 shows the difference between the diffserv bits and precedence bits in an IPv4 ToS byte and an IPv6 Traffic Class byte. Note that:

- Precedence bits are a subset of the Differentiated Services bits.
- The rightmost two bits are reserved.

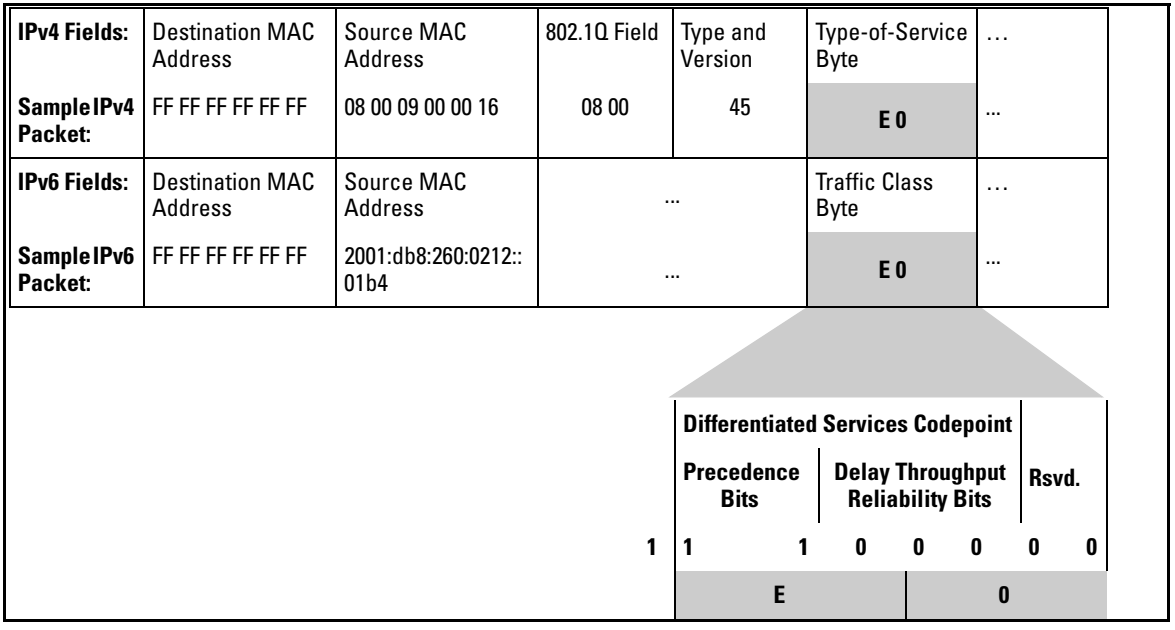


Figure 6-13. IPv4 ToS/IPv6 Traffic Class Byte with DSCP Codepoint and Precedence Bits

Assigning an 802.1p Priority for a Global IP-Precedence Classifier

If a device or application upstream of the switch sets the precedence bits in the ToS/Traffic Class byte of IPv4/IPv6 packets, you can use this global packet-marking option to prioritize packets for outbound port queues. If the outbound packets are in a tagged VLAN, this priority is carried as an 802.1p value to the adjacent downstream devices.

Syntax: qos type-of-service ip-precedence

Causes the switch to automatically assign an 802.1p priority to all IP packets (IPv4 and IPv6) by computing a packet's 802.1p priority from the precedence bits the packet carries. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (ToS IP Precedence Default: Disabled)

no qos type-of-service

Disables all ToS classifier operation, including prioritization using the precedence bits.

show qos type-of-service

When the IP-precedence mode is enabled (or if neither Type-of-Service option is configured), displays the ToS configuration status. If the Diff-serv mode is enabled, codepoint data is displayed as described in "Assigning a DSCP Policy for a Global IP-Diffserv Classifier" on page 6-49.

Using the IP-precedence classifier, prioritization of outbound packets relies on the IP-Precedence bit setting that IP packets carry with them from upstream devices and applications. To configure and verify this option:

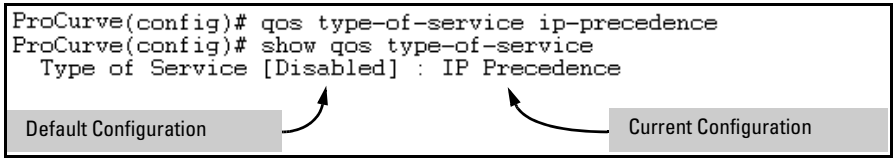


Figure 6-14. Example of Enabling ToS IP-Precedence Prioritization

To change from IP-precedence to IP-Diffserv mode, follow the procedure in "Assigning an 802.1p Priority for a Global IP-Diffserv Classifier" on page 6-45, which automatically disables IP-Precedence. To disable IP-Precedence without enabling the IP-Diffserv option, enter the **no qos type-of-service** command.

Assigning an 802.1p Priority for a Global IP-Diffserv Classifier

One of the best uses for this global QoS packet-marking option is on an interior switch where you want to honor (continue) a policy set on an edge switch. The IP-diffserv classifier enables you to select incoming packets having a specific DSCP and forward these packets with the desired 802.1p priority. For example, if an edge switch "A" marks all packets received on port A5 with a particular DSCP, you can configure a downstream (interior) switch "B" to handle such packets with the desired priority (regardless of whether 802.1Q-tagged VLANs are in use).

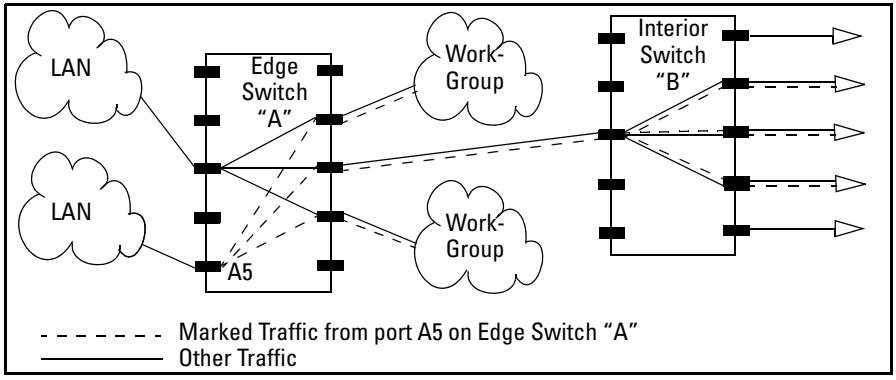


Figure 6-15. Interior Switch "B" Honors the Policy Established in Edge Switch "A"

To do so, assign the desired 802.1p priority to the same codepoint that the upstream or edge switch assigns to the selected packets. When the downstream switch receives an IP packet carrying one of these codepoints, it assigns the configured priority to the packet and sends it out the appropriate outbound port queue. (The packet retains the codepoint it received from the upstream or edge switch). You can use this option concurrently with the diffserv DSCP Policy option (described later in this section), as long as the DSCPs specified in the two options do not match.

Configuration Notes

Different applications may use the same DSCP in their IP packets. Also, the same application may use multiple DSCPs if the application originates on different clients, servers, or other devices. Using an edge switch enables you to select the desired packets and mark them with predictable DSCPs that can be used by downstream switches to honor policies set in the edge switch.

When enabled, the switch applies direct 802.1p prioritization to all packets having codepoints that meet these prerequisites:

- The codepoint is configured with an 802.1p priority in the DSCP table. (Codepoints configured with **No-override** are not used.)
- The codepoint is not configured for a new DSCP policy assignment.

Thus, the switch does not allow the same incoming codepoint (DSCP) to be used simultaneously for directly assigning an 802.1p priority and also assigning a DSCP policy. For a given incoming codepoint, if you configure one option and then the other, the second overwrites the first.

To use a global IP-Diffserv classifier to mark matching packets with an 802.1p priority, follow these steps:

1. Identify a DSCP used to set a policy in packets received from an upstream or edge switch.
2. Determine the 802.1p priority (0 - 7) you want to apply to packets carrying the identified DSCP. (You can either maintain the priority assigned in the upstream or edge switch, or assign a new priority.)
3. If necessary, use the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.
4. Enable IP-Diffserv mode by entering the **qos type-of-service diff-services** command.

.

Syntax: qos type-of-service diff-services < *codepoint* >

Causes the switch to read the **< *codepoint* >** (*DSCP) of an incoming IP packet and, when a match occurs, assign the associated 802.1p priority in the DSCP Policy table (see Table 6-11).*

no qos type-of-service

Disables all ToS classifier operation.

no qos dscp-map < *codepoint* >

Disables direct 802.1p priority assignment to packets carrying the < ***codepoint*** > *by reconfiguring the codepoint priority assignment in the DSCP table to* **No-override**. *Note that if this codepoint is in use as a DSCP policy for another diffserv codepoint, you must disable or redirect the other diffserv codepoint's DSCP policy before you can disable or change the codepoint. For example, in Figure 6-16 you cannot change the priority for the 000000 codepoint until you redirect the DSCP policy for 000001 from using 000000 as a policy. (See "Notes on Changing a Priority Setting" on page 6-91 and "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88.)*

show qos type-of-service

Displays the current Type-of-Service configuration. In IP-diffserv mode it also shows the current direct 802.1p assignments and the current DSCP assignments covered later in this section.

For example, an edge switch "A" in an untagged VLAN assigns a DSCP of 000110 on IP packets it receives on port A6, and handles the packets with high priority (7). When these packets reach interior switch "B" you want the switch to handle them with the same high priority. To enable this operation you would

configure an 802.1p priority of 7 for packets received with a DSCP of **000110**, and then enable **diff-services**:

```
ProCurve(config)# show qos type-of-service
 Type of Service [Disabled] : Disabled
 Codepoint DSCP Policy | Priority
 --------- ----------- + -----------
 000000                | 1
 000001    000000      | 1
 000010                | No-override
 000011                | No-override
 000100    001001      | 5
 000101                | No-override
 000110                | No-override
 000111                | No-override
 001000                | No-override
 001001                | 5
 001010                | 1
 001011                | No-override
   .          .          .
   .          .          .
   .          .          .
```

Executing this command displays the current Type-of-Service configuration and shows that the DSCP 000110 is not currently in use.

The **000110** codepoint is unused, and thus available for directly assigning an 802.1p priority without changing the packet's DSCP.

Note: All codepoints without a "DSCP Policy" entry are available for direct 802.1p priority assignment.

Figure 6-16. Displaying the Codepoints Available for 802.1p Priority Assignments

```
ProCurve(config)# qos dscp-map 000110 priority 7
ProCurve(config)# qos type-of-service diff-services

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : Differentiated Services
  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | 1
  000001    000000      | 1
  000010                | No-override
  000011                | No-override
  000100    001001      | 5
  000101                | No-override
  000110                | 7
  000111                | No-override
  001000                | No-override
  001001                | 5
    .          .          .
    .          .          .
    .          .          .
```

Outbound IP packets with a DSCP of **000110** will have a priority of **7**.

Notice that codepoints **000000** and **001001** are named as DSCP policies by other codepoints (**000001** and **000110** respectively). This means they are not available for changing to a different 802.1p priority.

Figure 6-17. Type-of-Service Configuration that Enables Both 802.1p Priority and DSCP Policy Assignment

Assigning a DSCP Policy for a Global IP-Diffserv Classifier

The preceding section describes how to forward an 802.1p priority level set by an edge (or upstream) switch. This section describes how to use a global IP-Diffserv classifier to mark matching packets with a new DSCP policy. A DSCP policy consists of a DSCP codepoint and an associated 802.1p priority.

You can use a global IP-Diffserv classifier to mark a DSCP policy at the same time with a global IP-Diffserv classifier that marks an 802.1p priority if different DSCP codepoints are configured with each classifier.

To use a global IP-Diffserv classifier to mark matching packets with a new DSCP policy, follow these steps:

1. Identify the DSCP used to set a policy in packets received from an upstream or edge switch.
2. Create a new policy by using the **qos dscp-map < *codepoint* > priority < 0 - 7>** command to configure an 802.1p priority for the codepoint you will use to overwrite the DSCP that the packet carries from upstream. (For more information, see “Differentiated Services Codepoint (DSCP) Mapping” on page 6-88.)
3. Use the **qos type-of-service diff-services < *incoming-DSCP* > dscp < *outgoing-DSCP* >** command to change the policy on packets coming from the edge or upstream switch with the specified incoming DSCP.

   Figure 6-15 on page 6-45 illustrates this scenario.

.

Syntax: qos type-of-service diff-services

Enables ToS diff-services.

Syntax: qos type-of-service diff-services < *current-codepoint* > dscp < *new-codepoint* >

Configures the switch to select an incoming IP packet carrying the <***current-codepoint***> *and then use the* <***new-codepoint***> *to assign a new, previously configured DSCP policy to the packet. The policy overwrites the* <***current-codepoint***> *with the* < ***new-codepoint***> *and assigns the 802.1p priority specified by the policy.*

Valid values for a DSCP codepoint are as follows:
- A binary value for the six-bit codepoint from **000000** *to* **111111**.
- A decimal value from **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
-An ASCII standard (hexadecimal) name for a binary DSCP bit set

Type **?** *to display the list of valid codepoint entries.*

To reconfigure the 802.1p priority currently assigned to a DSCP codepoint, use the **qos dscp-map** *command as described in**"Differentiated Services Codepoint (DSCP) Mapping" on page 6-88.*

Syntax: no qos type-of-service

Disables all ToS classifier operation. Current ToS DSCP policies and priorities remain in the configuration and will become available if you re-enable ToS diff-services.

Syntax: no qos type-of-service [diff-services < *codepoint* >]

Deletes the DSCP policy assigned to the < ***codepoint***> *and returns the* < ***codepoint***> *to the 802.1p priority setting it had before the DSCP policy was assigned. (This will be either a value from 0 - 7 or* **No-override***.)*

Syntax: show qos type-of-service

Displays a listing of codepoints with any corresponding DSCP policy re-assignments for outbound packets. Also displays the 802.1p priority for each codepoint that does not have a DSCP policy assigned to it.

Example. The following example shows how to configure new DSCP policies on matching packets with the specified DSCP codepoints.

Received DSCP	Policy DSCP	802.1p Priority	Policy Name (Optional)
001100	000010	6	Level 6
001101	000101	4	Level 4

1. Determine if the DSCP codepoints that you want to use to mark matching packets already have an 802.1p priority assigned, which could indicate use by existing applications (**show qos dscp-map** command). This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP (see "Notes on Changing a Priority Setting" on page 6-91).

   Also, note that a DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows **No-override** in the Priority column of the DSCP Policy table (**show qos dscp-map** command), you must first configure a priority for the codepoint before proceeding (**qos dscp-map priority** command). See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88 for more information.

```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
    .           .
    .           .
    .           .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 6-18. Displaying the Current DSCP-Map Configuration

2. Configure the desired policies (codepoint and associated 802.1p priority) in the DSCP table:

```
ProCurve(config)# qos dscp-map 000010 priority 6 name 'Level 6'
ProCurve(config)# qos dscp-map 000101 priority 4 name 'Level 4'

ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- --------------------------------
  000000      No-override
  000001      No-override
  000010      6           Level 6
  000011      No-override
  000100      No-override
  000101      4           Level 4
  000110      No-override
  000111      No-override
     .           .           .
     .           .           .
     .           .           .
```

Figure 6-19. Configuring DSCP Policies in the DSCP Table

3. Assign the new policies to mark matching packets with the specified codepoints.

```
ProCurve(config)# qos type-of-service diff-services 001100 dscp 000010
ProCurve(config)# qos type-of-service diff-services 001101 dscp 000101

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : Differentiated Services
  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | No-override
  000001                | No-override
  000010                | 6
  000011                | No-override
  000100                | No-override
  000101                | 4
  000110                | No-override
  000111                | No-override
  001000                | No-override
  001001                | No-override
  001010                | 1
  001011                | No-override
  001100    000010      | 6
  001101    000101      | 4
  001110                | 2
  001111                | No-override
  010000                | No-override
  010001                | No-override
-- MORE --, next page: Space, next line: Enter, quit: Control-C
```

The specified DSCP policies overwrite the original DSCPs in matching packets, and use the 802.1p priorities configured in the DSCP policies in step 2.

Figure 6-20. Assigning DSCP Policies to Outbound Packets Based on the DSCP Codepoint from Upstream Devices

Comparison of Global IP Type-of-Service Classifiers

Table 6-9 shows the difference in how global IP-Precedence and IP-Diffserv classifiers are implemented in the switch.

Table 6-9. IP Type-of-Service Classifiers

Outbound Port	IP Type-of-Service Classifiers	
	IP-Precedence Mode	**IP Differentiated Services Mode**
IP Packet Sent Out an Untagged Port in a VLAN	Based on the IP Precedence bit set in a packet's ToS/Traffic Class field, the packet is sent to one of eight outbound port queues in the switch: 1 - 2 = low priority (queue 1, 2) 0 - 3 = normal priority (queue 3, 4) 4 - 5 = medium priority (queue 5, 6) 6 - 7 = high priority (queue 7, 8)	Based on the DSCP codepoint that the switch has been configured to detect, one of the following actions is taken: • The codepoint is re-marked according to the configured DSCP policy and the 802.1p priority currently configured for the codepoint in the DSCP Policy table (Table 6-8). • The codepoint is not changed, but the 802.1p priority is marked with the currently configured value for the codepoint in the DSCP Policy table. Based on the new 802.1p priority marking, the packet leaves the switch through one of the following queues: 1 - 2 = low priority (queue 1, 2) 0 - 3 = normal priority (queue 3, 4) 4 - 5 = medium priority (queue 5, 6) 6 - 7 = high priority (queue 7, 8) If **No-override** (the default) is configured for the 802.1p priority associated with a codepoint, the priority in the packet header is not re-marked by the global IP-Diffserv classifier and, by default, is sent to the "normal priority" outbound port queue.
IP Packet Sent Out a Tagged Port in a VLAN	Based on the IP Precedence bit set in a packet's ToS/Traffic Class field: • The packet is sent to one of eight outbound port queues in the switch as described above. • The IP Precedence value (0 - 7) is used to set the corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device (see Table 6-8).	Based on the DSCP codepoint that the switch has been configured to detect, one of the following actions is taken: • The codepoint is re-marked according to the configured DSCP policy and the 802.1p priority currently configured for the codepoint in the DSCP Policy Table (Table 6-8). • The codepoint is not changed, but the 802.1p priority is marked with the currently configured value for the codepoint in the DSCP Policy Table (Table 6-8). Based on the new 802.1p priority marking, the packet leaves the switch through one of the outbound port queues described above. In addition, the priority value (0 - 7) is used to set the 802.1p priority in the VLAN tag carried by the packet to the next downstream device. If the priority is configured as **No-override** in the DSCP Policy table, the VLAN tag carries a "0" (normal priority) 802.1p setting if not prioritized by other global QoS classifiers.

Global Layer-3 Protocol Classifier

Global QoS Classifier Precedence: 4

When a global Layer-3 Protocol classifier is configured as the highest-precedence classifier and the switch receives traffic carrying the specified protocol, matching packets are assigned the priority configured for the classifier. (For information on QoS operation when other global QoS classifiers match the same traffic, see "Globally-Configured Packet Classification" on page 6-12.)

Assigning a Priority for a Global Layer-3 Protocol Classifier

This global QoS packet-marking option assigns an 802.1p priority to outbound packets having the specified Layer-3 protocol.

Syntax: qos protocol

< ip | ipx | arp | appletalk | sna | netbeui > priority < 0 - 7 >

Configures an 802.1p priority for outbound packets having the specified protocol. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each protocol type. (Default: **No-override***)*

no qos protocol

< ip | ipx | arp | appletalk | sna | netbeui >

Disables use of the specified protocol as a QoS classifier and resets the protocol priority to **No-override**.

show qos protocol

Lists the QoS protocol classifiers with their priority settings.

Example. In this example, you configure the following global Layer-3 protocol classifiers:

1. Configure QoS protocol classifiers with IP at 0 (normal), ARP at 5 (medium), and AppleTalk at 7 (high) and display the QoS protocol configuration.
2. Disable the QoS IP protocol classifier, downgrade the ARP priority to 4, and again display the QoS protocol configuration.

Figure 6-21 shows the configuration commands.

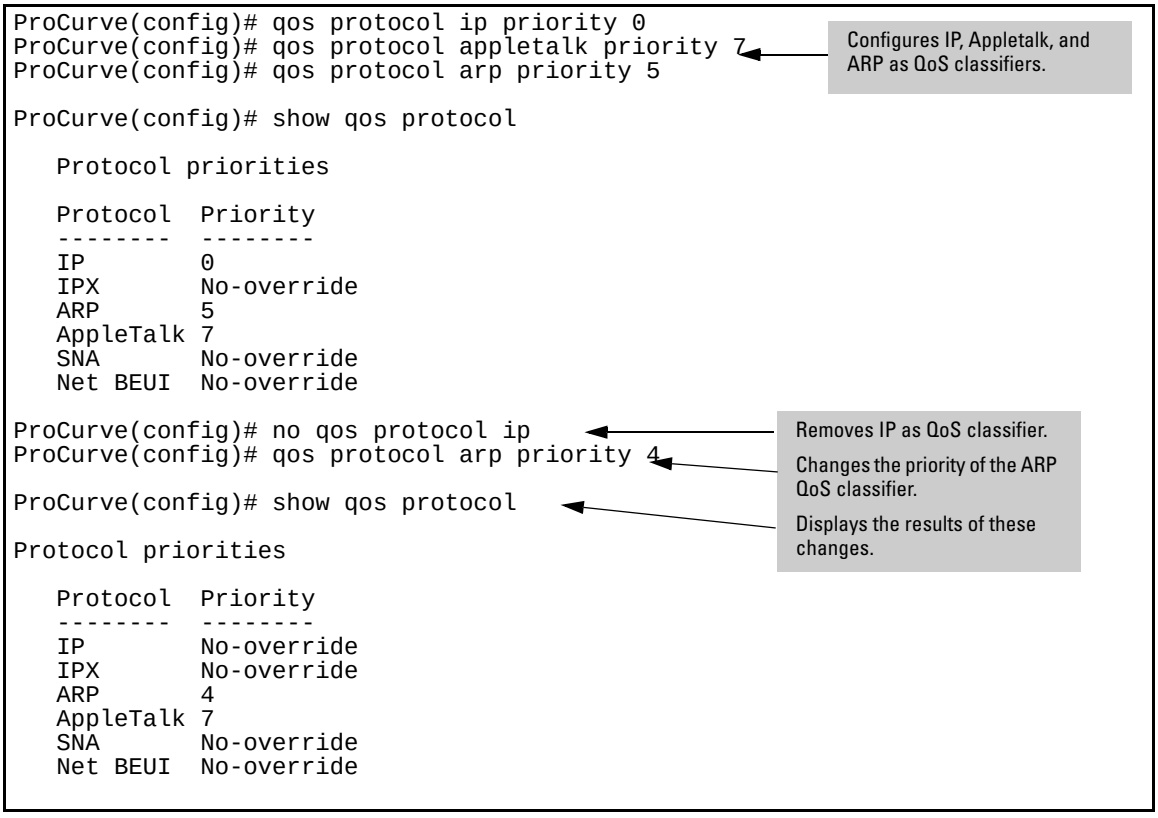


Figure 6-21. Adding, Displaying, Removing, and Changing QoS Protocol Classifiers

Global VLAN-ID Classifier

Global QoS Classifier Precedence: 5

The global VLAN-ID (VID) classifier allows you to use up to 4094 VLAN IDs to match packets. When a particular VLAN-ID classifier has the highest precedence in the switch, traffic received in the VLAN is marked with the configured priority level. You can configure different global VLAN-ID classifiers to mark packets with different priority levels.

Options for Assigning Priority. The global QoS packet-marking options for packets that carry a specified VLAN-ID include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

For information on QoS operation when other global QoS classifiers match the same traffic, see to "Globally-Configured Packet Classification" on page 6-12.

Configuration Notes

A global VLAN-ID classifier marks priority levels only in packets received on static VLANs. Packets received in a dynamic VLAN created by GVRP operation are not marked by a global VLAN-ID classifier.

The VLAN ID used as a global QoS classifier must currently exist on the switch. If you remove a VLAN from the switch, all global QoS configurations that use the VLAN ID for packet marking are also removed.

Assigning a Priority for a Global VLAN-ID Classifier

This global QoS packet-marking option assigns a priority to all outbound packets having the specified VLAN-ID (VID). You can configure this option by either specifying the VLAN-ID ahead of the **qos** command or moving to the VLAN context for the VLAN you want to configure for priority.

Syntax: vlan < *vid* > qos priority < 0 - 7 >

Configures an 802.1p priority for outbound packets belonging to the specified VLAN. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each VLAN-ID. (Default: **No-override***)*

Syntax: no vlan < *vid* > qos

Removes the specified VLAN-ID as a QoS classifier and resets the priority for that VLAN to **No-override**.

Syntax: show qos vlan-priority

Displays a listing of the QoS VLAN-ID classifiers currently in the running-config file, with their priority data.

Example. In this example, 802.1p priorities are assigned to packets received in VLANs 1, 20, 30, and 40.

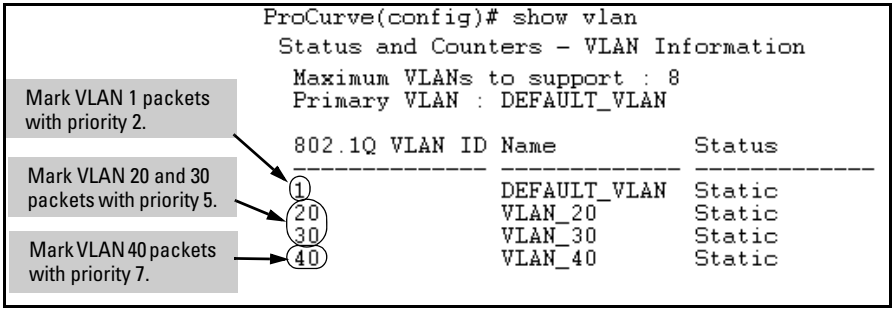

ProCurve(config)# show vlan
Status and Counters - VLAN Information
Maximum VLANs to support : 8
Primary VLAN : DEFAULT_VLAN
802.1Q VLAN ID Name Status
1 DEFAULT_VLAN Static
20 VLAN_20 Static
30 VLAN_30 Static
40 VLAN_40 Static
Mark VLAN 1 packets with priority 2.
Mark VLAN 20 and 30 packets with priority 5.
Mark VLAN 40 packets with priority 7.

Figure 6-22. Displaying the VLANs Available for QoS Prioritization

Enter the following commands to mark VLAN packets that match the specified VLAN IDs with an 802.1p priority:

```
ProCurve(config)# vlan 1 qos priority 2
ProCurve(config)# vlan 20 qos priority 5
ProCurve(config)# vlan 30 qos priority 5
ProCurve(config)# vlan 40 qos priority 7

ProCurve(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       Priority    |        2
  20      Priority    |        5
  30      Priority    |        5
  40      Priority    |        7
```

Figure 6-23. Configuring and Displaying QoS Priorities for VLAN IDs

If you later decided to remove VLAN 20 from QoS prioritization, you would enter the following command:

```
ProCurve(config)# no vlan 20 qos
ProCurve(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule   | DSCP   Priority
  ------- ------------ + ------ ------------
  1       Priority     |        2
  20      No-override  |        No-override
  30      Priority     |        5
  40      Priority     |        7
```

In this instance, **No- override** indicates that VLAN 20 is not prioritized by QoS.

Figure 6-24. Returning a QoS-Prioritized VLAN to "No-override" Status

Assigning a DSCP Policy for a Global VLAN-ID Classifier

This global QoS packet-marking option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified VLAN-ID (VID). The switch:

1. Selects an incoming IP packet on the basis of the VLAN-ID it carries.
2. Overwrites the packet's DSCP with the DSCP configured for matching packets.
3. Assigns the 802.1p priority associated with the new DSCP. (See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88.)
4. Forwards the packet through the appropriate outbound port queue.

Creating a Policy Based on VLAN-ID Classifier.

1. Determine the VLAN-ID classifier to which you want to assign a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected VLAN-ID:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.

3. If necessary, use the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.

Prerequisite

A DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows **No-override** in the Priority column of the DSCP Policy table (**show qos dscp-map** command), you must first configure a priority for the codepoint before proceeding (**qos dscp-map priority** command). See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88 for more information.

4. Configure the switch to assign the DSCP policy to packets with the specified VLAN-ID.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

This command is optional if a priority has already been assigned to the < ***codepoint*** >*. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this priority to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IP packets, the DSCP codepoint in the packet header is replaced by the codepoint specified in this command. (Default: For most codepoints,* **No-override**. *See Figure 6-11 on page 6-89.)*

Syntax: vlan < *vid* > qos dscp < *codepoint* >

Assigns a DSCP policy to IP packets carrying the specified VLAN ID, and overwrites the DSCP in these packets with the assigned < ***codepoint*** > *value.*

Valid values for a DSCP codepoint are as follows:
- A binary value for the six-bit codepoint from **000000** *to* **111111**.
- A decimal value from **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
-An ASCII standard name for a binary DSCP bit set

Type **?** *to display the list of valid codepoint entries.*

The DSCP policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: **No-override***)*

Syntax: no vlan < *vid* > qos

Removes a global QoS classifier for the specified VLAN.

Syntax: show qos device-priority

Displays a listing of all QoS VLAN-ID classifiers currently in the running-config file.

Example. This example, assigns the following DSCP policies (codepoint and associated 802.1p priority) to packets with the specified VLAN IDs:

VLAN-ID	DSCP	Priority
40	000111	7
30	000101	5
20	000010	1
1	000010	1

1. Determine if the DSCP codepoints that you want to use to mark matching packets already have an 802.1p priority assigned, which could indicate use by existing applications (**show qos dscp-map** command). This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP (see "Notes on Changing a Priority Setting" on page 6-91.

   Note that a DSCP codepoint must also have a priority configured before you can use it to mark matching packets.

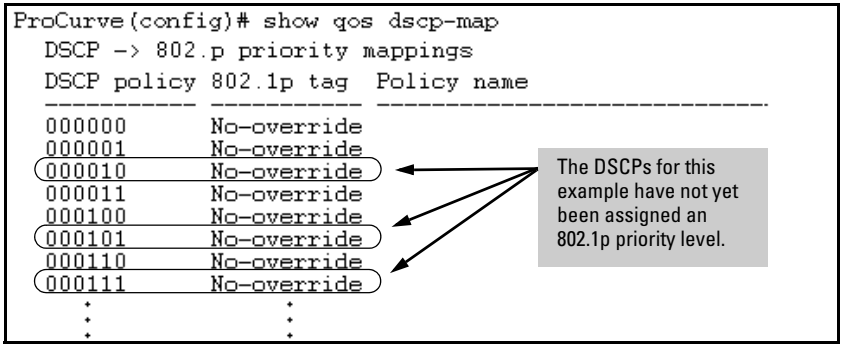


Figure 6-25. Displaying the Current DSCP-Priority Mapping in the DSCP Policy Table

2. Configure the priorities for the DSCPs you want to use.

```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
    .           .
    .           .
    .           .
```

802.1p priorities are configured in this step.

Figure 6-26. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected VLAN IDs and display the result.

```
ProCurve(config)# vlan 1 qos dscp 000010
ProCurve(config)# vlan 20 qos dscp 000010
ProCurve(config)# vlan 30 qos dscp 000101
ProCurve(config)# vlan 40 qos dscp 000111

ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       DSCP        | 000010 1
  20      DSCP        | 000010 1
  30      DSCP        | 000101 5
  40      DSCP        | 000111 7
```

Figure 6-27. The Completed VID-DSCP Priority Configuration

The switch will now apply the DSCP policies in figure 6-27 to packets received on the switch with the specified VLAN-IDs. This means the switch will:

- Overwrite the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assign the 802.1p priorities in the above policies to the appropriate packets.

Global Source-Port Classifier

Global QoS Classifier Precedence: 6

The global QoS source-port classifier allows you to use a packet's source-port on the switch to mark packets. When a global source-port classifier has the highest precedence in the switch for traffic entering through a port, traffic received on the port is marked with the configured priority level. Different source-port classifiers can have different priority levels.

Options for Assigning Priority on the Switch. The global QoS packet-marking options for matching packets from a specified source-port include:

- 802.1p priority
- DSCP policy: Assigning a new DSCP and an associated 802.1p priority

For information on QoS operation when other global QoS classifiers match the same traffic, see to "Globally-Configured Packet Classification" on page 6-12.

Options for Assigning Priority From a RADIUS Server. You can use a RADIUS server to assign a QoS source-port priority during an 802.1X port-access authentication session. Refer to the RADIUS chapter in the *Access Security Guide* for your switch.

Assigning a Priority for a Global Source-Port Classifier

This global QoS packet-marking option assigns a priority to all outbound packets having the specified source-port. You can configure this option by either specifying the source-port ahead of the **qos** command or moving to the port context for the port you want to configure for priority. (If you are configuring multiple source-ports with the same priority, you may find it easier to use the **interface < *port-list* >** command to go to the port context instead of individually configuring the priority for each port.)

Syntax: interface < *port-list* > qos priority < 0 - 7 >

Configures an 802.1p priority for packets entering the switch through the specified (source) ports. This priority determines the packet queue in the outbound port(s) to which traffic is sent. If a packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each source-port or group of source-ports. (Default: **No-override***)*

Syntax: no interface < *port-list* > qos

Disables use of the specified source-port(s) for QoS classifier(s) and resets the priority for the specified source-port(s) to **No-override**.

Syntax: show qos port-priority

Lists the QoS port-priority classifiers with their priority data.

Example. This example shows how to prioritize inbound traffic on the following source-ports:

Source-Port	Priority
A1 - A3	2
A4	3
B1, B4	5
C1-C3	6

Enter the following commands to prioritize packets received from the specified source ports:

:

```
ProCurve(config)# interface e c1-c3 qos priority 6
ProCurve(config)# interface e b1,b4 qos priority 5
ProCurve(config)# interface e a4 qos priority 3
ProCurve(config)# interface e a1-a3 qos priority 2

ProCurve(config)# show qos port-priority
  Port priorities
  Port Apply rule   | DSCP   Priority    Radius Override
  ---- ------------ + ------ ----------- ---------------
  A1   Priority     |        2           No-override
  A2   Priority     |        2           No-override
  A3   Priority     |        2           No-override
  A4   Priority     |        3           No-override
  B1   Priority     |        5           No-override
  B2   No-override  |        No-override No-override
  B3   No-override  |        No-override No-override
  B4   Priority     |        5           No-override
  C1   Priority     |        6           No-override
  C2   Priority     |        6           No-override
  C3   Priority     |        6           No-override
  C4   No-override  |        No-override No-override
  C5   No-override  |        No-override No-override
   .         .                   .
   .         .                   .
   .         .                   .
```

Figure 6-28. Configuring and Displaying Source-Port QoS Priorities

If you later decided to remove source-port A1 from QoS prioritization, you would enter the following command:

```
ProCurve(config)# no interface e a1 qos
ProCurve(config)# show qos port-priority
  Port priorities

  Port Apply rule   | DSCP   Priority     Radius Override
  ---- ------------ + ------ ------------ ---------------
  A1   No-override  |        No-override  No-override
  A2   Priority     |        2            No-override
  A3   Priority     |        2            No-override
  A4   Priority     |        3            No-override
```

In this instance, **No-override** indicates that port A1 is not prioritized by QoS.

Figure 6-29. Returning a QoS-Prioritized VLAN to "No-override" Status

Assigning a DSCP Policy for a Global Source-Port Classifier

This global QoS packet-marking option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets received from the specified source-ports. The switch:

1. Selects an incoming IP packet on the basis of its source-port.
2. Overwrites the packet's DSCP with the DSCP configured for matching packets.
3. Assigns the 802.1p priority associated with the new DSCP. (See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88.)
4. Forwards the packet through the appropriate outbound port queue.

For more on DSCP, refer to "QoS Terminology" on page 6-7.

Creating a Policy Based on Source-Port Classifiers.

Configuration Notes

You can configure only one DSCP per source-port to mark matching packets.

Configuring a new DSCP for a source-port automatically overwrites (replaces) any previous DSCP or 802.1p priority configuration for that source-port classifier.

1. Identify the source-port classifier to which you want to assign a DSCP policy.
2. Determine the DSCP policy for packets having the selected source-port:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received through the source-port from upstream devices.)

b. Determine the 802.1p priority you want to assign to the DSCP.

3. If necessary, use the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.

Prerequisite

A DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows **No-override** in the Priority column of the DSCP Policy table (**show qos dscp-map** command), you must first configure a priority for the codepoint before proceeding (**qos dscp-map priority** command). See "Differentiated Services Codepoint (DSCP) Mapping" on page 6-88 for more information.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

This command is optional if a priority has already been assigned to the < ***codepoint***>. *The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this priority to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IP packets, the DSCP codepoint in the packet header is replaced by the codepoint specified in this command. (Default: For most codepoints,* **No-override**. *See Figure 6-11 on page 6-89.)*

4. Configure the switch to assign the DSCP policy to packets from the specified source-port.

Syntax: interface < *port-list* > qos dscp < *codepoint* >

Assigns a DSCP policy to IP packets from the specified source-port(s), and overwrites the DSCP in these packets with the assigned < ***codepoint***> *value.*

Valid values for a DSCP codepoint are as follows:
- A binary value for the six-bit codepoint from **000000** *to* **111111**.
- A decimal value from **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
-An ASCII standard name for a binary DSCP bit set
Type **?** *to display the list of valid codepoint entries.*

Syntax: interface < *port-list* > qos dscp < *codepoint* >

The DSCP policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: **No-override***)*

Syntax: no interface [e] < *port-list* > qos

Removes a QoS classifier for the specified source-port(s).

Syntax: show qos source-port

Displays a listing of all source-port QoS classifiers currently in the running-config file.

Example. In this example, you assign the following DSCP policies (codepoint and associated 802.1p priority) to matching packets with the specified source-ports:

Source-Port	DSCP	Priority
A2	000111	7
B1-B3	000101	5
B4, C2	000010	1

1. Determine if the DSCP codepoints that you want to use to mark matching packets already have an 802.1p priority assigned, which could indicate use by existing applications (**show qos dscp-map** command). This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP.

   Also, note that a DSCP must have an 802.1p priority configured before you can use it to mark matching packets. If necessary, use the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.

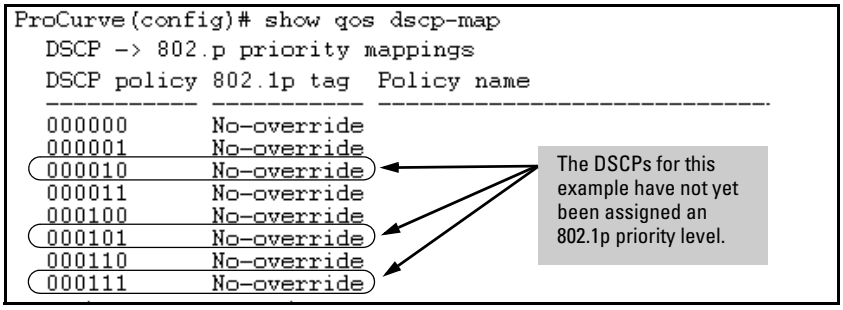


Figure 6-30. Displaying the Current DSCP-Priority Mapping in the DSCP Policy Table

2. Configure the priorities for the DSCPs that you want to use to mark matching packets.

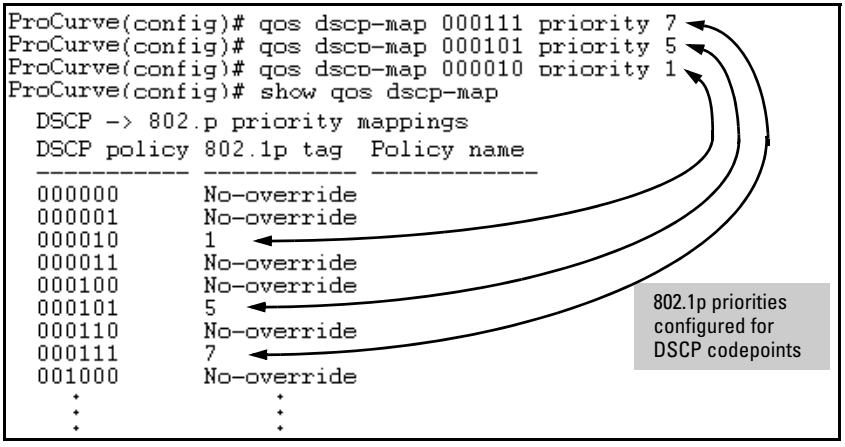


Figure 6-31. Assigning Priorities to the Specified DSCP Codepoints

3. Assign the DSCP policies to the selected source-ports and display the result.

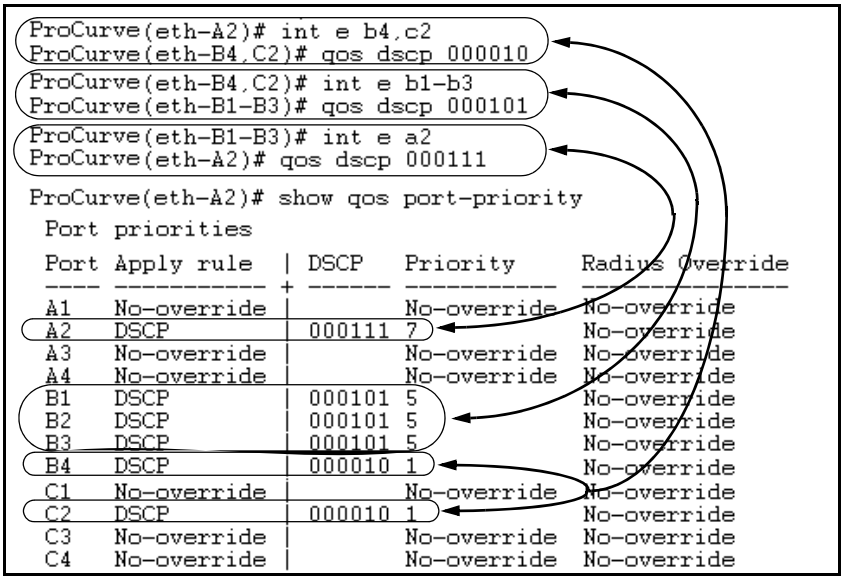

ProCurve(eth-A2)# int e b4,c2
ProCurve(eth-B4,C2)# qos dscp 000010
ProCurve(eth-B4,C2)# int e b1-b3
ProCurve(eth-B1-B3)# qos dscp 000101
ProCurve(eth-B1-B3)# int e a2
ProCurve(eth-A2)# qos dscp 000111
ProCurve(eth-A2)# show qos port-priority
Port priorities
Port Apply rule | DSCP Priority Radius Override
A1 No-override | No-override No-override
A2 DSCP | 000111 7 No-override
A3 No-override | No-override No-override
A4 No-override | No-override No-override
B1 DSCP | 000101 5 No-override
B2 DSCP | 000101 5 No-override
B3 DSCP | 000101 5 No-override
B4 DSCP | 000010 1 No-override
C1 No-override | No-override No-override
C2 DSCP | 000010 1 No-override
C3 No-override | No-override No-override
C4 No-override | No-override No-override

Figure 6-32. Global Source-Port Classifier with DSCP-Priority Marking

Radius Override Field. During a client session authenticated by a RADIUS server, the server can impose a port priority that applies only to that client session. Refer to the RADIUS chapter in the *Access Security Guide* for your switch.

IP Multicast (IGMP) Interaction with QoS

IGMP high-priority-forward causes the switch to service the subscribed IP multicast group traffic at high priority, even if QoS on the switch has relegated the traffic to a lower priority. This does not affect any QoS priority settings, so the QoS priority is honored by downstream devices. However, QoS does take precedence over IGMP normal-priority traffic.

The switch's ability to prioritize IGMP traffic for either a normal or high priority outbound queue overrides any QoS criteria, and does not affect any 802.1p priority settings the switch may assign. For a given packet, if both IGMP high priority and QoS are configured, the QoS classification occurs and the switch marks the packet for downstream devices, but the packet is serviced by the high-priority queue when leaving the switch.

IGMP High Priority	QoS Configuration Affects Packet	Switch Port Output Queue	Outbound 802.1p Setting (Requires Tagged VLAN)
Not Enabled	Yes	Determined by QoS	Determined by QoS
Enabled	See above paragraph.	High	As determined by QoS if QoS is active.

Advanced Classifier-Based QoS

Starting in software release K.14.01, in addition to the packet classification and prioritization methods described in "Globally-Configured QoS" on page 6-19, QoS configuration also supports advanced classifier-based functions. Advanced classifier-based QoS introduces:

- A finer granularity than globally-configured QoS for classifying IPv4 and IPv6 traffic
- Additional actions for managing selected traffic, such as rate-limiting and IP precedence marking
- The application of QoS policies to inbound traffic flows on specific port and VLAN interfaces (instead of using only globally-configured, switch-wide QoS settings)
- The ability to re-use traffic classes in different software-feature configurations, such as QoS and port mirroring

Classifier-based QoS is designed to work with existing globally-configured, switch-wide QoS policies by allowing you to zoom in on a subset of port or VLAN traffic to further manage it. Classifier-based policies take precedence over, and may override, globally-configured QoS settings that apply to all traffic on the switch.

Classifier-based QoS policies provide greater control for managing network traffic. Using multiple match criteria, you can finely select and define the classes of traffic that you want to manage. QoS-specific policy actions determine how you can handle the selected traffic.

For more information, refer to the "Classifier-Based Software Configuration" chapter in the *Advanced Traffic Management Guide*.

Classifier-Based QoS Model

Classifier-based QoS configuration consists of the following general steps:

1. Classify the traffic that you want to manage by configuring a class.
2. Configure a QoS policy in which you specify the QoS actions to execute on each class of traffic.
3. Assign the QoS policy to a port or VLAN (inbound only) interface.

Note

Classifier-based QoS operation supports all globally-configured packet classification criteria (except for Source-port and Layer-3 protocol) and traffic marking functions, and provides additional QoS actions on a per-port and per-VLAN basis.

- Classifier-based match criteria on inbound IPv4/IPv6 traffic include:
  - IP source address (IPv4 and IPv6)
  - IP destination address (IPv4 and IPv6)
  - IP protocol (such as ICMP or SNMP)
  - Layer 3 IP precedence bits
  - Layer 3 DSCP codepoint
  - Layer 4 TCP/UDP application port (including TCP flags)
  - VLAN ID
- Classifier-based QoS policy actions on matching IPv4/IPv6 packets are as follows:
  - Setting the Layer 2 802.1p priority value (class of service) in VLAN-tagged and untagged packet headers
  - Setting the Layer 3 IP precedence bits
  - Setting the Layer 3 Differentiated Services Codepoint (DSCP) bits
  - Rate-limiting inbound traffic on port and VLAN interfaces

Classifier-Based QoS Configuration Procedure

To use the classifier-based model to configure a QoS policy and apply it to a selected class of traffic on a port or VLAN interface, follow these steps:

1. Evaluate the types of traffic in your network and identify the traffic types that you want to prioritize or rate-limit.

2. Create an IPv4 or IPv6 traffic class using the **class** command to select the packets you want to manage.

Context: Global configuration

Syntax: [no] class < ipv4 | ipv6 > <*classname* >

Defines the name of a traffic class and specifies whether a policy is to be applied to IPv4 or IPv6 packets, where < classname > is a text string (64 characters maximum). After you enter the **class** *command, you enter the class configuration context to specify match criteria. A traffic class contains a series of* **match** *and* **ignore** *commands, which specify the criteria used to classify packets.*

A traffic class consists of match criteria, which consist of **match** and **ignore** commands.

- **match** commands define the values that header fields must contain for a packet to belong to the class and be managed by policy actions.
- **ignore** commands define the values which, if contained in header fields, exclude a packet from the policy actions configured for the class.

Note

Be sure to enter match/ignore statements in the *precise order* in which you want their criteria to be used to check packets.

The following match criteria are supported in match/ignore statements for inbound IPv4/IPv6 traffic:

- IP source address (IPv4 and IPv6)
- IP destination address (IPv4 and IPv6)
- Layer 2 802.1Q VLAN ID
- Layer 3 IP protocol
- Layer 3 IP precedence bits
- Layer 3 DSCP codepoint
- Layer 4 TCP/UDP application port
- VLAN ID

Enter one or more **match** or **ignore** commands from the class configuration context to filter traffic and determine the packets on which policy actions will be performed.

Context: Class configuration

Syntax: [no] [*seq-number*] < match | ignore > < *ip-protocol* >
< *source-address* > < *destination-address* > [dscp *codepoint*]
[precedence *precedence-value*] [tos *tos-value*] [vlan *vlan-id*]

For detailed information about how to enter **match** and **ignore** commands to configure a traffic class, refer to the "Creating a Traffic Class" section in the "Classifier-Based Software Configuration" chapter.

3. Create a QoS policy to perform QoS actions on selected packets by entering the **policy qos** command from the global configuration context.

Context: Global configuration

Syntax: [no] policy qos <*policy-name* >

Defines the name of a QoS policy and enters the policy configuration context.

A traffic policy consists of one or more classes, and one or more QoS actions configured for each class of traffic. The configured actions are executed on packets that match a **match** statement in a class. No policy action is performed on packets that match an **ignore** statement.

Note

Be sure to enter each class and its associated QoS actions in the *precise order* in which you want packets to be checked and processed by QoS actions.

To configure the QoS actions that you want to execute on packets that match the criteria in a specified class, enter one or more **class action** commands from the policy configuration context:

Context: Policy configuration

Syntax: [no] [*seq-number*] class < ipv4 | ipv6 > <*classname* >
action <*qos-action* > [action <*qos-action* > ...]

Defines the QoS actions to be applied on a pre-configured IPv4 or IPv6 traffic class when a packet matches the **match** *criteria in the traffic class. You can enter multiple action statements for the same traffic class.*

[no] [*seq-number*] class < ipv4 | ipv6 > <*classname* >

- [***seq-number***] — *The (optional)* ***seq-number*** *parameter sequentially orders the QoS actions that you enter in a policy configuration. Actions are executed on matching packets in numerical order. Default: QoS action statements are numbered in increments of 10, starting at 10.*
- **class** < **ipv4** | **ipv6** > <*classname* > — *Defines the preconfigured traffic class on which the QoS actions in the policy are executed, and specifies whether the QoS policy is applied to IPv4 or IPv6 traffic in the class. The classname is a text string (64 characters maximum).*

Note*: You can configure multiple* **class action** *statements for different traffic classes in the same policy. The execution of QoS actions is performed in the order in which the actions are numerically listed in the policy.*

action <*qos-action* > [action <*qos-action* > ...]

The **action** *keyword configures the QoS action specified by the* ***qos-action*** *parameter. The action is executed on any packet that matches the* **match** *criteria in the class. The action is not executed on packets that match* **ignore** *criteria.*

The complete **no** *form of the* **class action** *command or the* **no <*seq-number* >** *command removes a QoS action from the policy configuration.*

The following QoS commands are supported by the ***qos-action*** *parameter:*

- **rate-limit** <*kbps* **>**
- **priority <** *priority-value* **>**
- **ip-precedence** < *precedence-value* **>**
- **dscp <** *dscp-value*

For information on the complete syntax of each QoS command, see "Configuring QoS Actions in a Policy" on page 6-76.

To manage packets that do not match the **match** or **ignore** criteria in any class in the policy, and therefore have no QoS actions performed on them, you can enter an optional default class. The default class is placed at the end of a policy configuration and specifies the QoS actions to perform on packets that are neither matched nor ignored.

4. (Optional) To configure a default class in a policy, enter the **default-class** command at the end of a policy configuration and specify one or more QoS actions to be executed on packets that are not matched and not ignored.

Context: Policy configuration

Syntax: [no] default-class action <*qos-action*> [action <*qos-action*> ...]

Configures a default class that allows one or more QoS actions to be executed on packets that are not matched or ignored by any of the class configurations in a QoS policy. The default-class supports the same QoS commands as the **class** < **ipv4** | **ipv6** > **action** *command:* **rate-limit**, **priority**, **ip-precedence**, *and* **dscp**.

For general information about how to configure and manage a service policy, refer to the "Creating a Service Policy" section in the "Classifier-Based Software Configuration" chapter.

5. Apply the QoS policy to inbound traffic on a port (**interface service-policy in** command) or VLAN (**vlan service-policy in** command) interface.

The following restrictions apply to a QoS service policy:

- Only one QoS policy is supported on a port or VLAN interface.
- If you apply a QoS policy to a port or VLAN interface on which a QoS policy is already configured, an error message is displayed. The new policy does not overwrite the existing one. To apply a new policy, you must first remove the existing policy with the **no policy** command.
- A QoS policy is supported only on inbound traffic.

Because only one QoS policy is supported on a port or VLAN interface, ensure that the policy you want to apply contains all the required classes and actions for your configuration.

To apply a QoS policy on a port or VLAN interface, enter one of the following commands from the global configuration context.

Context: Global configuration

Syntax: interface <*port-list*> service-policy <*policy-name*> in

Configures the specified port(s) with a QoS policy that is applied to inbound traffic on each interface. Separate individual port numbers in a series with a comma; for example, **a1, b4, d3**. *Enter a range of ports by using a dash; for example,* **a1-a5**. *The QoS policy name you enter must be the same as the policy name you configured with the* **policy qos** *command in Step 2.*

Syntax: vlan <*vlan-id*> service-policy <*policy-name*> in

Configures a QoS policy on the specified VLAN that is applied to inbound traffic on the VLAN interface.
Valid VLAN ID numbers range from 1 to 4094.
The QoS policy name you enter must be the same as the policy name you configured with the **policy** *command in Step 2.*

For more information about how to apply a QoS policy to an interface, refer to the "Applying a Service Policy to an Interface" section in the "Classifier-Based Software Configuration" chapter.

6. Determine the additional QoS configurations that you need to apply to each QoS-capable device in your network and configure the appropriate policy.

   (Optional) If you want 802.1p (CoS) priority settings to be included in outbound packets, configure tagged VLANs on the appropriate downstream links.

Configuring QoS Actions in a Policy

In QoS policy-configuration mode, you define the actions to be applied to a pre-configured IPv4 or IPv6 traffic class when a packet matches the **match** criteria in the class. (Actions are not executed on packets that match **ignore** criteria.) You can enter multiple action statements in a traffic class, including the default class.

The following actions are supported in a QoS policy configuration:

- **rate-limit** command: Configures the rate-limit for matching packets.
- **ip-precedence** command: Configures (marks) the IP precedence bits in the ToS byte of IPv4 packet headers and Traffic Class byte of IPv6 headers.
- **dscp** command: Configures the DSCP bits in the IPv4 ToS byte and IPv6 Traffic Class byte of packet headers.
- **priority** command: Configures the 802.1p class of service (CoS) priority in Layer 2 frame headers.

For information on the difference between the DSCP bits and precedence bits in the ToS byte of an IPv4 header and the Traffic Class byte of an IPv6 header, see Figure 6-13.

Context: Global configuration

Syntax: [no] [*seq-number*] class < ipv4 | ipv6 > <*classname* >
action <***qos-action*** > [**action** <***qos-action*** > ...]

In a QoS policy configuration, the ***qos-action*** *parameter can be any of the following commands:*

- **rate-limit** <*kbps*>**:** *Configures the maximum transmission rate for matching packets in a specified traffic class. All packets that exceed the configured limit are dropped.*

  The rate limit is specified in kilobits per second, where **<** *kbps* **>** *is a value from 0 to 10000000.*

  Rate-Limiting Usage Notes:

  – *Rate-limit values below 13 kbps may result in unpredictable rate-limiting behavior.*

  – *Configuring a rate limit of 0 (zero) kilobits on a port blocks all traffic on the port. If blocking all traffic is the desired behavior, ProCurve recommends that you configure a* **deny** *ACL instead configuring a rate-limit of 0.*

  – *A rate-limit that you apply with a classifier-based policy overrides any globally-configured per-port rate-limit on the selected packets.*

  For more information on per-port rate-limiting, refer to the "Port Traffic Controls" chapter in the Management and Configuration Guide.

  Rate-Limiting Restrictions:

  – *A rate-limit is calculated on a per-module or per port-bank basis. If trunked ports or VLANs with a configured rate-limit span multiple modules or port-banks, the configured rate-limit is not guaranteed.*

  – *A QoS policy that uses the* **class action rate-limit** *command is not supported on a port interface on which ICMP rate-limiting has already been globally configured. To apply the QoS policy, you must first disable the ICMP rate-limiting configuration.*

  In cases where you want to maintain an ICMP rate-limiting configuration, configure a class in which you specify the necessary match statements for ICMP traffic, and a QoS policy in which you configure the rate-limit action for the class.

action <*qos-action*> [action <*qos-action*> ...]
—Continued—

For information on globally-configured ICMP, refer to the "Configuring ICMP" section in the "Configuring IP Parameters for Routing Switches" chapter in the Multicast and Routing Guide.

- **priority <** *priority-value* **>:** *Configures the 802.1p class of service (CoS) bits in Layer 2 frames of matching packets in a specified traffic class. Valid CoS values range from 0 to 7.*

  The 802.1p CoS value controls the outbound port-queue priority for traffic leaving the switch. In an 802.1Q VLAN network, downstream devices may honor or change the 802.1p priority in incoming packets. For more information, see "Layer 2 802.1p Prioritization" on page 6-14.

  Table 6-3 shows how the Layer 2 802.1p priority value determines to which outbound port queue a packet is sent both on the switch and on a downstream device.

  The 802.1p CoS numeric value (from 0 to 7) corresponds to the hexadecimal equivalent of the three binary "0" and "1" bit settings in the Layer 2 header. For example if the CoS bit values are **1 1 1**, *the numeric value is* **7** *(1+2+4). Similarly, if the CoS bits are* **0 1 1**, *the numeric value is* **3** *(1+2+0).*

  Note: *If you want the 802.1p CoS priority settings included in outbound packets to be honored on downstream devices, configure tagged VLANs on the appropriate inbound and outbound ports.*

- **ip-precedence** < *precedence-value* **>:** *Configures the IP precedence value in the IPv4 ToS byte or IPv6 Traffic Class byte of matching packets in a specified traffic class. Valid IP precedence values are either a numeric value from 0 (low priority) to 7 (high priority) or its corresponding name:*

  0 routine
  1 priority
  2 immediate
  3 flash
  4 flash-override
  5 critical
  6 internet (for internetwork control)
  7 network (for network control)

action <*qos-action*> [action <*qos-action*> ...]
—*Continued*—

Table 6-2 shows how the Layer 2 802.1p priority value determines to which outbound port queue a packet is sent.

Table 6-8 shows the 802.1p priority value (0 to 7) associated, by default, with each IP Precedence three-bit setting and automatically assigned by the switch to the Layer 2 header of matching packets.

- **dscp <** *dscp-value* **>:** *Configures the DSCP codepoint in the IPv4 ToS byte or IPv6 Traffic Class byte of matching packets in a specified traffic class.*

  Valid values for the DSCP codepoint are any of the following:

  – *A binary eight-bit set (such as* **100110***)*
  – *A decimal value from* **0** *(low priority) to* **63** *(high priority) that corresponds to a binary DSCP bit set*
  – *The ASCII standard name for a binary DSCP bit set:*

  af11 (001010) **af42** (100100)
  af12 (001100) **af43** (100110)
  af13 (001110) **ef** (101110)
  af21 (010010) **cs1** (001000) = *precedence 1*
  af22 (010100) **cs2** (010000) = *precedence 2*
  af23 (010110) **cs3** (011000) = *precedence 3*
  af31 (011010) **cs4** (100000) = *precedence 4*
  af32 (011100) **cs5** (101000) = *precedence 5*
  af33 (011110) **cs6** (110000) = *precedence 6*
  af41 (100010) **cs7** (111000) = *precedence 7*
  default (000000)

  Prerequisite*: The DSCP value you enter must already be configured with an 802.1p priority in the DSCP Policy table (Table 6-11) before you can use it to mark matching packets.*

DSCP-802.1p Mapping

The 802.1p priority currently associated with each DSCP codepoint is stored in the DSCP Policy table (displayed with the **show qos dscp-map** command and shown in Table 6-11). Note that certain DSCP codepoints have 802.1p priorities assigned by default. The 802.1p priority mapped to a DSCP codepoint is automatically applied in matching packets whose codepoint is reset with the **class action dscp** command in a QoS policy.

To reconfigure the 802.1p priority value currently assigned to a DSCP codepoint, you can enter either of the following commands:

- **qos dscp-map** <*codepoint*> **priority** <0-7> (global configuration context)
- **class** <**ipv4** | **ipv6**> <*classname*> **action dscp** <*codepoint*> **priority** <0-7> (policy configuration context)

If you do not enter a **priority** value with the **class action dscp** command in a QoS policy, one of the following occurs:

- The switch refers to the DSCP Policy table to assign the 802.1p value that is currently configured for the specified DSCP codepoint to remark matching packets.
- If the specified DSCP codepoint is not associated with an 802.1p priority in the DSCP Policy table, an error message is displayed and the **class action dscp** <*codepoint*> command is not executed. You are prompted to re-enter the command with an 802.1p priority: **class action dscp** <*codepoint*> **priority** <0-7>.

To ensure that the desired 802.1p priority is assigned to matching packets, you may need to first re-map the priority to the new codepoint before you configure the policy, by using the **qos dscp-map** <*codepoint*> **priority** <0-7> command.

Note that after you reconfigure the 802.1p priority for a DSCP codepoint, the switch immediately applies the new 802.1p priority value to packets transmitted with the associated codepoint as a result of:

- Globally-configured QoS commands
- **class action dscp** commands in other QoS policies

Example. In the following example, a classifier-based QoS policy (**dscp-remap**) that assigns a new DSCP codepoint (**af43**) and associated 802.1p priority (**5**) to matching packets with a specified DSCP codepoint (**af11**) is applied to the inbound traffic on a VLAN.

```
ProCurve(config)# qos dscp-map af43 priority 5
ProCurve(config)# class ipv4 dscp5
ProCurve(config-class)# match ip any any dscp af11
ProCurve(config-class)# exit
ProCurve(config)# policy qos dscp-remap
ProCurve(config-policy)# class ipv4 dscp5 action dscp af43
ProCurve(config-policy)# exit
ProCurve# vlan 3 service-policy dscp-remap in
```

Note that in the preceding example, the desired 802.1p priority is mapped to the specified DSCP codepoint by using the **qos dscp-map** <*codepoint*> **priority** <0-7> command *before* the QoS policy is configured.

Override of Global QoS Settings

After you apply a QoS policy to an interface, the classifier-based settings configured by QoS actions in the policy override any 802.1p CoS or DSCP codepoint values that were globally-configured on the switch to mark packets using the QoS commands described in "Globally-Configured QoS" on page 6-19.

If you use a classifier-based QoS configuration along with globally-configured QoS commands, the order of precedence in which 802.1p priority, IP precedence, and DSCP settings mark selected packets is as follows, from highest (1) to lowest (9):

Table 6-10. Order of Precedence for Classifier-Based QoS over Global QoS

Precedence Order	QoS Feature	Reference
1	Classifier-based port-specific policy	Page 6-71
2	Classifier-based VLAN-specific policy	Page 6-71
3	Globally-configured TCP/UDP priority	Page 6-24
4	Globally-configured IP-device priority	Page 6-33
5	Globally-configured IP Type-of-Service priority	Page 6-41
6	Globally-configured Layer 3-Protocol priority	Page 6-54
7	Globally-configured VLAN-ID priority	Page 6-56
8	Globally-configured Source-Port priority	Page 6-62
9	802.1p CoS in Layer 2 VLAN header[1]	Page 6-12

[1] In a tagged VLAN environment, the incoming 802.1p priority is used as the default QoS classifier to determine how a packet is handled if no global or classifier-based QoS match criterion with a higher precedence matches.

Viewing a Classifier-Based QoS Configuration

Use the following **show** commands to display information about a classifier-based QoS configuration and statistics or resource usage on QoS policies.

Syntax: show class < *class-name* >

Displays a traffic-class configuration, including all match/ignore statements used to classify packets.

```
ProCurve# show class ipv4 gnutella
Statements for Class ipv4 "gnutella"
  10 match tcp 0.0.0.0 255.255.255.255 range 6346 6347 0.0.0.0 255.255.255.255
  20 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 range 6346 6347
  30 match udp 0.0.0.0 255.255.255.255 range 6346 6347 0.0.0.0 255.255.255.255
  40 match udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 range 6346 6347

ProCurve# show class ipv4 kazaa
Statements for Class ipv4 "kazaa"
  10 match tcp 0.0.0.0 255.255.255.255 eq 1214 0.0.0.0 255.255.255.255
  20 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 1214
  30 match udp 0.0.0.0 255.255.255.255 eq 1214 0.0.0.0 255.255.255.255
  40 match udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 1214

ProCurve# show class ipv4 http
Statements for Class ipv4 "http"
  10 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 80
  20 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 443
  50 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 8080
```

Figure 6-33. Example of "show class" Output for a QoS Policy

Syntax: show policy < *policy-name* >

Displays a policy configuration, including the traffic classes and QoS commands to be executed on each class.

```
ProCurve# show policy suspect-traffic
Statements for Policy "suspect-traffic"
   10 class ipv4 "http" action rate-limit kbps 2000 action priority 3
   20 class ipv4 "kazaa" action rate-limit kbps 1000 action priority 2
   30 class ipv4 "gnutella" action rate-limit kbps 1000 action priority 2
```

Figure 6-34. Example of "show policy" Output for a QoS Policy

Syntax: show statistics policy [*policy-name*] [port <*port-list*> | vlan <*vlan-id*>] in

Displays the statistics for a specified QoS policy or all QoS policies configured on the switch, a specified port, or a VLAN interface for inbound traffic. Separate individual port numbers in a series with a comma; for example, **a1, b4, d3**. *Enter a range of ports by using a dash; for example,* **a1-a5**. *Valid VLAN ID numbers range from 1 to 4094.*

```
ProCurve# show statistics policy suspect-traffic vlan 300 in

HitCounts for Policy suspect-traffic

10 class ipv4 "http" action rate-limit kbps 2000 action priority 3 [ Meter 975000
kilo bits]
(150)   10 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 80
(0)     20 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 443
(200)   30 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 8000
(0)     40 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 8001
(300)   50 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 8080

20 class ipv4 "kazaa" action rate-limit kbps 1000 action priority 2 [ Meter 0 kilo
bits]
(0)     10 match tcp 0.0.0.0 255.255.255.255 eq 1214 0.0.0.0 255.255.255.255
(0)     20 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 1214
(0)     30 match udp 0.0.0.0 255.255.255.255 eq 1214 0.0.0.0 255.255.255.255
(0)     40 match udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 1214

30 class ipv4 "gnutella" action rate-limit kbps 1000 action priority 2 [ Meter 0
kilo bits]
(0)   10 match tcp 0.0.0.0 255.255.255.255 range 6346 6347 0.0.0.0 255.255.255.255
(0)   20 match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 range 6346 6347
(0)   30 match udp 0.0.0.0 255.255.255.255 range 6346 6347 0.0.0.0 255.255.255.255
(0)   40 match udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 range 6346 6347
```

Number of packets (in parentheses) that have matched the criteria in the match/ignore statement in each class in the QoS policy and have been processed by the action configured for the class

Figure 6-35. Example of "show statistics policy" Output for a QoS Policy

Syntax: show policy resources

Displays the number of hardware resources (rules, meters, and application port ranges) used by classifier-based QoS policies that are currently applied to interfaces on the switch as well as mirroring policies and other software features.

Note*: The information displayed is the same as the output of the* **show qos resources** *(see Figure 6-8) and* **show access-list resources** *commands.*

For a detailed explanation of the information displayed with the **show <qos | access-list | policy> resources** *command, refer to the "Monitoring Resources" appendix of the Management and Configuration Guide.*

```
ProCurve# show policy resources

 Resource usage in Policy Enforcement Engine
        |    Rules    |  Rules Used
  Ports |  Available  |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+-------------+-------+-------+-------+-------+--------+-------|
  1-24  |        3014 |    15 |    11 |     0 |     1 |      0 |     3 |
  25-48 |        3005 |    15 |    10 |    10 |     1 |      0 |     3 |
  A     |        3017 |    15 |     8 |     0 |     1 |      0 |     3 |

        |    Meters   |  Meters Used
  Ports |  Available  |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+-------------+-------+-------+-------+-------+--------+-------|
  1-24  |         250 |       |     5 |     0 |       |        |     0 |
  25-48 |         251 |       |     4 |     0 |       |        |     0 |
  A     |         253 |       |     2 |     0 |       |        |     0 |

        | Application |
        | Port Ranges |  Application Port Ranges Used
  Ports |  Available  |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+-------------+-------+-------+-------+-------+--------+-------|
  1-24  |        3014 |     2 |     0 |     0 |       |      0 |     0 |
  25-48 |        3005 |     2 |     0 |     0 |       |      0 |     0 |
  A     |        3017 |     2 |     0 |     0 |       |      0 |     0 |

 0 of 8 Policy Engine management resources used.
 Key:
 ACL = Access Control Lists
 QoS = Device & Application Port Priority, QoS Policies, ICMP rate limits
 IDM = Identity Driven Management
 VT  = Virus Throttling blocks
 Mirror = Mirror Policies, Remote Intelligent Mirror endpoints
 Other = Management VLAN, DHCP Snooping, ARP Protection, Jumbo IP-MTU.

 Resource usage includes resources actually in use, or reserved for future
 use by the listed feature.  Internal dedicated-purpose resources, such as
 port bandwidth limits or VLAN QoS priority, are not included.
```

Includes the hardware resources used by classifier-based QoS policies that are currently applied to inerfaces on the switch.

Figure 6-36. Example of "show policy resources" Output for all Currently Configured QoS Policies

Classifier-Based QoS Restrictions

The following restrictions apply to QoS policies configured with the classifier-based model:

- You cannot apply a classifier-based QoS policy on a port or VLAN interface on which a classifier-based QoS policy is already configured. You can, however, apply a classifier-based policy of a different type, such as port mirroring.
- A QoS policy that uses the **rate-limit** command is not supported on a port interface on which ICMP rate-limiting has already been globally configured. To apply the QoS policy, you must first disable the ICMP rate-limiting configuration. For more information, refer to the "Configuring ICMP" section in the "Configuring IP Parameters for Routing Switches" chapter in the *Multicast and Routing Guide*.

  In cases where you want to maintain an ICMP rate-limiting configuration, configure a QoS policy in which you add the necessary **match** statements for the ICMP traffic in a class configuration, and then configure a **rate-limit** action for the class in the policy configuration.
- In a QoS policy that uses the **class action rate-limit** command, the rate-limit is calculated on a per-module or per port-bank basis. If trunked ports or VLANs with a configured rate-limit span multiple modules or port-banks, the configured rate-limit is not guaranteed.
- In a QoS policy that uses the **class action dscp** command, the DSCP value you enter must be already configured with an 802.1p priority in the DSCP Policy table (see Table 6-11).

Interaction with Other Software Features

After you apply a QoS policy to an interface, you may receive an error message if there are not sufficient hardware resources to support the policy. In this case, use the **show resources** command to verify the amount of resources that are currently in use and the resources available on the switch. QoS policies share the same hardware resources with other software features, such as mirroring policies, ACLs, virus throttling, the management VLAN, and so on.

Use the displayed information to decide if you need to re-prioritize current resource usage by reconfiguring or disabling software features to free the resources reserved for less important features.

For more information, refer to "Displaying Resource Usage for QoS Policies" on page 6-32 and the "Monitoring Resources" chapter in the *Management and Configuration Guide*.

Classifier-Based QoS Configuration Examples

QoS Policy for Voice over IP and Data Traffic

In this example, an administrator would like to configure the following Layer 2 802.1p CoS and Layer 3 DSCP values to prioritize how VoIP traffic from different phones is handled compared to data traffic:

- Softphone traffic: DSCP 46; 802.1p CoS priority 6
- Avaya phone traffic: DSCP 34; 802.1p CoS priority 3
- Miscellaneous phone traffic: DSCP 26; 802.1p CoS priority 3
- Data traffic: DSCP 000000; 802.1p CoS priority 0

The following QoS configuration creates and assigns a QoS policy to VLAN 1 that prioritizes VoIP and data traffic in this way:

```
ProCurve# class ipv4 DataTraffic
ProCurve(config-class)# match ip any any dscp 0
ProCurve(config-class)# exit
ProCurve# class ipv4 softphoneTraffic
ProCurve(config-class)# match tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
ip-dscp 46
ProCurve(config-class)# exit
ProCurve# class ipv4 DigiPhoneTraffic
ProCurve(config-class)# match ip 12.255.100.10/24 any ip-dscp 34
ProCurve(config-class)# match ip 10.255.100.12/24 any ip-dscp 26
ProCurve(config-class)# exit
ProCurve# policy qos prioritizeVoIP
ProCurve(config-policy)# class ipv4 DataTraffic action priority 0
ProCurve(config-policy)# class ipv4 softphoneTraffic action priority 6
ProCurve(config-policy)# class ipv4 DigiPhoneTraffic action priority 3
ProCurve(config-policy)# exit
ProCurve# vlan 2 service-policy prioritizeVoIP in
```

These match statements select traffic that satisfies multiple criteria; for example, a TCP port range and a DSCP value or a source IP address and a DSCP value.

Figure 6-37. Example of a QoS Policy for Voice over IP and Data Traffic

QoS Policy for Layer 4 TCP/UDP Traffic

The following example shows how to configure a rate-limiting policy for TCP/UDP application streams and apply the policy on all inbound switch ports.

```
ProCurve# class ipv4 http
ProCurve(config-class)# match tcp any any eq 80
ProCurve(config-class)# match tcp any any eq 443
ProCurve(config-class)# match tcp any any eq 8080
ProCurve(config-class)# exit
ProCurve# class ipv4 kazaa
ProCurve(config-class)# match tcp any eq 1214 any
ProCurve(config-class)# match tcp any any eq 1214
ProCurve(config-class)# exit
ProCurve# class ipv4 gnutella
ProCurve(config-class)# match tcp any range 6346 6347 any
ProCurve(config-class)# match tcp any any range 6346 6347
ProCurve(config-class)# match udp any range 6346 6347 any
ProCurve(config-class)# match udp any any range 6346 6347
ProCurve(config-class)# exit
ProCurve# policy qos PrioritizeSuspectTraffic
ProCurve(config-policy)# class ipv4 http action rate-limit kbps 7000
ProCurve(config-policy)# class ipv4 kazaa action rate-limit kbps 3500
ProCurve(config-policy)# class ipv4 gnutella action rate-limit kbps 3500
ProCurve(config-policy)# exit
ProCurve# interface all service-policy PrioritizeSuspectTraffic in
```

Figure 6-38. Example of a QoS Policy for TCP/UDP Applications on Inbound Port Interfaces

QoS Policy for Subnet Traffic

The next example shows how to configure a QoS policy that prioritizes inbound traffic sent to and received from a specified subnet (15.29.16.0/10) and TCP port range on VLAN 5.

```
ProCurve# class ipv4 adminTraffic
ProCurve(config-class)# match ip 15.29.16.1/10 any
ProCurve(config-class)# match ip any 15.29.16.1/10
ProCurve(config-class)# match tcp ::/0 ::/0 range 100 200 ip-dscp 46
ProCurve(config-class)# exit
ProCurve# policy prioritizeAdminTraffic
ProCurve(config-policy)# class ipv4 adminTraffic action priority 7
ProCurve(config-policy)# exit
ProCurve# vlan 5 service-policy prioritizeAdminTraffic in
```

Match statement with IPv6 source and destination addresses.

Figure 6-39. Example of a QoS Policy for IPv4 and IPv6 Subnet Traffic on a VLAN Interface

Differentiated Services Codepoint (DSCP) Mapping

The DSCP Policy Table associates an 802.1p priority with a DSCP codepoint in an IPv4/IPv6 packet. Using DSCP codepoints in your network allows you to set a LAN policy that operates independently of 802.1Q VLAN-tagging.

In the default state, most of the 64 codepoints do not assign an 802.1p priority, as indicated by **No-override** in Table 6-11 on page 6-89. However, some codepoints, such as Assured Forwarding and Expedited Forwarding, have a default 802.1p priority setting.

Use the following commands to display the DSCP Policy table, configure the codepoint-priority assignments, and assign optional names to the codepoints.

Syntax: show qos dscp-map

Displays the DSCP Policy table.

qos dscp-map **< *codepoint* >** priority < 0 - 7 > [name **< *ascii-string* >**]

Configures an 802.1p priority for the specified codepoint and an optional (DSCP policy) name.

no qos dscp-map **< *codepoint* >**

Removes the currently configured 802.1p priority that is associated with the specified **<*codepoint*>** *and displays the* **No-override** *setting. The codepoint policy name, if configured, is also removed.*

no qos dscp-map < *codepoint* > name

Deletes only the policy name, if configured, for the specified **< *codepoint* >**.

Table 6-11. The Default DSCP Policy Table

DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority
000000	No-override	010110	3*	101011	No-override
000001	No-override	010111	No-override	101100	No-override
000010	No-override	011000	No-override	101101	No-override
000011	No-override	011001	No-override	101110	7**
000100	No-override	011010	4*	101111	No-override
000101	No-override	011011	No-override	110000	No-override
000110	No-override	011100	4*	110001	No-override
000111	No-override	011101	No-override	110010	No-override
001000	No-override	011110	5*	110011	No-override
001001	No-override	011111	No-override	110100	No-override
001010	1*	100000	No-override	110101	No-override
001011	No-override	100001	No-override	110110	No-override
001100	1*	100010	6*	110111	No-override
001101	No-override	100011	No-override	111000	No-override
001110	2*	100100	6*	111001	No-override
001111	No-override	100101	No-override	111010	No-override
010000	No-override	100110	7*	111011	No-override
010001	No-override	100111	No-override	111100	No-override
010010	0 *	101000	No-override	111101	No-override
010011	No-override	101001	No-override	111110	No-override
010100	0 *	101010	No-override	111111	No-override
010101	No-override				

*Assured Forwarding codepoints; configured by default on the switches covered in this guide.
**Expedited Forwarding codepoint configured by default.

Default Priority Settings for Selected Codepoints

In a few cases, such as 001010 and 001100, a default DSCP policy (implied by the DSCP standards for Assured-Forwarding and Expedited-Forwarding) is used. You can change the priorities for the default policies by using the **qos dscp-map < *codepoint* > priority < 0 - 7 >** command.

The currently configured DSCP policies (codepoint and associated 802.1p priority) are not enabled until you configure a global or classifier-based QoS policy to mark matching packets or configure a global IP-Diffserv classifier.

Displaying Non-Default Codepoint Settings

Table 6-11 displays the switch's default codepoint-priority assignments. If you change the priority of any codepoint to a non-default value and then enter the **write memory** command, the switch will list the non-default setting in the **show config** display.

Example. the default configuration has the following DSCP-priority settings:

Codepoint	Default Priority
001100	1
001101	No-override
001110	2

If you reconfigure these three codepoints to a priority of 3 and then enter the **write memory** command, the switch displays the changes in the **show config** listing:

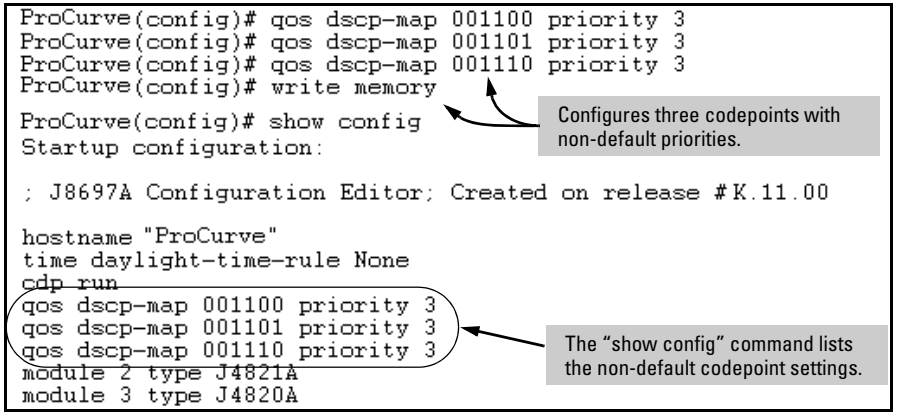

ProCurve(config)# qos dscp-map 001100 priority 3
ProCurve(config)# qos dscp-map 001101 priority 3
ProCurve(config)# qos dscp-map 001110 priority 3
ProCurve(config)# write memory
ProCurve(config)# show config
Startup configuration:
; J8697A Configuration Editor; Created on release #K.11.00
hostname "ProCurve"
time daylight-time-rule None
cdp run
qos dscp-map 001100 priority 3
qos dscp-map 001101 priority 3
qos dscp-map 001110 priority 3
module 2 type J4821A
module 3 type J4820A
Configures three codepoints with non-default priorities.
The "show config" command lists the non-default codepoint settings.

Figure 6-40. Displaying Non-Default Priority Settings in the DSCP Table

Effect of "No-override". If you configure a global IP-Diffserv classifier and **No-override** is displayed for the 802.1p priority associated with a codepoint, DSCP marking cannot be performed on matching outbound packets. However, QoS does not affect the packet-queuing 802.1p priority or VLAN tagging carried in the packet.

In this case, the packets are handled as follows (as long as no other QoS classifier marks a new 802.1p priority on the matching packets):

802.1Q Status	Outbound 802.1p Priority
Received and forwarded on a tagged-port member of a VLAN	Unchanged
Received on an untagged-port member of a VLAN; forwarded on a tagged-port member of a VLAN	0 (zero)—"normal"
Forwarded on an untagged-port member of a VLAN	None

Notes on Changing a Priority Setting

If you try to modify the priority associated with a DSCP codepoint in a DSCP policy using the **qos dscp-map priority** command, and if the DSCP policy is currently used by one or more global QoS or classifier-based QoS policies, the following error message is displayed:

```
Cannot modify DSCP Policy < codepoint > - in use by
other qos rules.
```

In this case, you can enter the following QoS **show** commands to identify in which global and classifier-based QoS configurations the DSCP policy is currently used:

show policy <*qos-policy*>
show qos tcp-udp-port-priority
show qos device-priority
show qos type-of-service
show qos protocol
show qos vlan
show qos port-priority

After determining the QoS configurations in which the DSCP-priority mapping is used, you can either delete a QoS configuration and reset the DSCP-priority mapping to **No-override**, or change either the 802.1p priority or the codepoint used in the QoS configuration.

Example. If codepoint 000001 is currently mapped to priority 6, and several global QoS policies use this codepoint to assign a priority to their respective types of matching traffic, you can change the priority associated with the codepoint as follows:

1. Identify the global and classifier-based QoS policies that use the codepoint.
2. Do one of the following:

a. Reconfigure each QoS policy by re-entering a different DSCP codepoint or a different 802.1p priority associated with the codepoint.

b. Enter the **no qos < *classifier* >** or **no policy < *qos-policy* >** command to remove the current DSCP policy with codepoint 000001 and reset the priority to **No-override**.

3. Use the **qos dscp-map 000001 priority < 0 - 7 >** command to remap DSCP 000001 to the desired priority.

4. Do one of the following:

   a. Reconfigure codepoint 000001 in the QoS policies in which you want to use the new DSCP-priority mapping to mark matching packets.

   b. Leave a QoS policy in which you use DSCP 000001 to mark matching packets with the associated **No-override** priority mapping.

Error Messages for DSCP Policy Changes

Refer to the error messages in Table 6-12 to troubleshoot an error condition that results from reconfiguring a DSCP policy.

Error Message	Description
DSCP Policy < ***decimal-codepoint*** > not configured	You have tried to configure a codepoint in a global or classifier-based QoS policy for which there is no associated priority (**No-override**). Use the **qos dscp-map** command to configure a priority for the codepoint, then re-enter the codepoint in the QoS configuration.
Cannot modify DSCP Policy < ***codepoint*** > - in use by other qos rules.	You have tried to configure a codepoint in a global or classifier-based QoS policy that is already in use by other QoS policies. Before remapping the codepoint to a new priority, you must first reconfigure the other QoS policies so that they do not use the current codepoint-priority mapping. You can have multiple QoS policies that use the same codepoint to mark packets as long as it is acceptable for all such policies to use the same 802.1p priority.

Table 6-12. Error Messages Generated by DSCP Policy Changes

Example of Changing the Priority Setting on a Policy When One or More Classifiers Are Currently Using the Policy

In this example, the codepoint 000001 is in use by one or more global QoS policies. If you try to modify the priority currently associated with the codepoint, an error message similar to the following is displayed:

```
ProCurve(config)# qos dscp-map 000001 priority 2
Cannot modify DSCP Policy 000001 - in use by other qos rules.
```

Figure 6-41. Error Message for Changing the Priority on a DSCP Policy

In this case, you would use steps similar to the following to change the priority.

1. Identify the global and classifier-based QoS policies that use the codepoint whose DSCP-priority mapping you want to change.

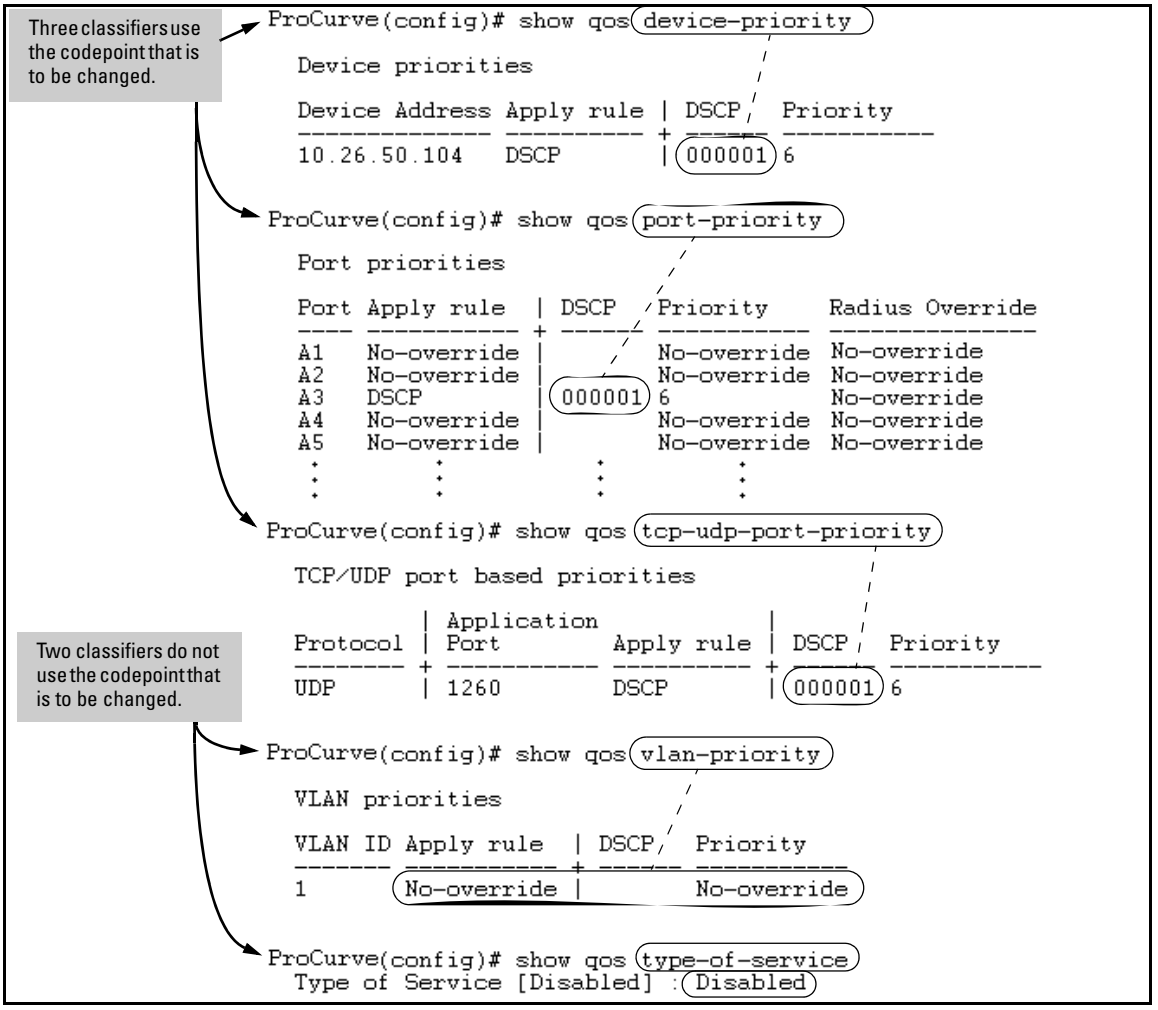

Three classifiers use the codepoint that is to be changed.
ProCurve(config)# show qos device-priority
Device priorities
Device Address Apply rule | DSCP Priority
10.26.50.104 DSCP | 000001 6
ProCurve(config)# show qos port-priority
Port priorities
Port Apply rule | DSCP Priority Radius Override
A1 No-override | No-override No-override
A2 No-override | No-override No-override
A3 DSCP | 000001 6 No-override
A4 No-override | No-override No-override
A5 No-override | No-override No-override
ProCurve(config)# show qos tcp-udp-port-priority
TCP/UDP port based priorities
Protocol | Application Port Apply rule | DSCP Priority
UDP | 1260 DSCP | 000001 6
Two classifiers do not use the codepoint that is to be changed.
ProCurve(config)# show qos vlan-priority
VLAN priorities
VLAN ID Apply rule | DSCP Priority
1 No-override | No-override
ProCurve(config)# show qos type-of-service
Type of Service [Disabled] : Disabled

Figure 6-42. Identifying the QoS Policies that Use a Codepoint

2. Change each QoS configuration by assigning a different DSCP policy or a different 802.1p priority, or by removing the currently configured DSCP policy and restore the default **No-override** setting; for example:
   a. Delete the current DSCP policy used to mark matching packets for a global IP-device policy (**no qos device-priority** command) and reset the default priority mapping to **No-override**.
   b. Create a new DSCP policy to use when you reconfigure QoS policies to use the new codepoint-priority mapping.
   c. Configure a global QoS source-port policy to mark matching packets with the new DSCP policy.
   d. Assign the global QoS policy that matches **udp-port 1260** packets to a different 802.1p priority.

```
(a) ProCurve(config)# no qos device-priority 10.26.50.104
(b) ProCurve(config)# qos dscp-map 000100 priority 6
(c) ProCurve(config)# int e a3 qos dscp 000100
(d) ProCurve(config)# qos udp-port 1260 priority 2
```

3. Reconfigure the desired priority for the 000001 codepoint.

```
ProCurve(config)# qos dscp-map 000001 priority 4
```

4. Reconfigure QoS policies with the original codepoint (000001) to mark packets with the new DSCP-priority mapping, or leave QoS policies as currently configured from Step 2.

QoS Queue Configuration

QoS queue configuration allows you to reduce the number of outbound queues that all switch ports use to buffer packets for 802.1p user priorities. By default:

- Series 3500yl switches use four queues.
- Series 5400zl switches and the Series 8212zl switch use eight queues.

You can change the default QoS queue configuration to four-queue mode or two-queue mode to increase the available bandwidth per queue.

Use the following commands to change the number of queues per port and display the current priority queue configuration on the switch.

Syntax: qos queue-config < 2-queues | 4-queues | 8-queues >

Configures the number of outbound priority queues for all ports on the switch using one of the following options: **2-queues**, **4-queues**, *or* **8-queues**.
(Default: 8-queues)

Caution: *This command will execute a "write memory" followed by an immediate reboot, replacing the Startup configuration with the content of the current Running configuration.*

The new configuration will:
1. Remove any previously configured "bandwidth-min output" settings
2. Set the new number of outbound port queues
If you select anything but "yes" for this operation, the operation is aborted and a message stating "Operation aborted" appears.

show qos queue config

Displays the current QoS queue configuration.

Mapping of Outbound Port Queues. The mapping of 802.1p priorities to outbound port queues is shown in Table 6-13.

Table 6-13. Mapping of 802.1p Priorities to Outbound Port Queues

802.1p Priority	8 Queues (default)	4 Queues	2 Queues
1 (lowest)	1	1	1
2	2		
0 (normal)	3	2	
3	4		
4	5	3	2
5	6		
6	7	4	
7 (highest)	8		

Impact of QoS Queue Configuration on Guaranteed Minimum Bandwidth (GMB). Changing the number of queues removes any **bandwidth-min output** settings in the startup configuration, and automatically re-allocates the GMB per queue as shown in Table 6-14.

Table 6-14. Default GMB Percentage Allocations per QoS Queue Configuration

802.1p Priority	8 Queues (default)		4 Queues		2 Queues	
	Queue	GMB	Queue	GMB	Queue	GMB
1 (lowest)	1	2%	1	8%	1	20%
2	2	3%				
0 (normal)	3	30%	2	17%		
3	4	10%				
4	5	10%	3	30%	2	80%
5	6	10%				
6	7	15%	4	45%		
7 (highest)	8	20%				

Note

For more information on configuring GMB, refer to the chapter titled "*Port Traffic Controls*" in the *Management and Configuration Guide*.

Configuring the Number of Priority Queues

To change the number of outbound priority queues for all ports on the switch, use the **qos queue-config** command.

Caution

The **qos queue-config** command executes a **write memory** followed by an immediate reboot, replacing the Startup configuration with the contents of the current Running configuration. In addition to setting the number of outbound port queues, the new configuration will remove any previously configured **bandwidth-min output** settings.

Example. To change the number of outbound priority queues for all ports on the switch from eight queues (the default) to four:

1. Configure the number of outbound priority queues by using the **qos queue-config** command.

   ```
   ProCurve(config)# qos queue-config 4-queues
   ```

   A caution message is displayed (see the Caution note above) concluding with the following prompt.

   ```
   Do you wish to proceed? [Proceed/Cancel]
   ```

2. Type **Proceed** to continue.

   A second confirmation prompt appears:

   ```
   Please confirm reset. [Yes/Cancel]
   ```

3. Type **Yes** to initiate a write memory followed by an immediate reboot. (If you enter **Cancel** at either of the two prompts, the command is aborted and the current queue configuration is maintained on the switch).

   The changes will be committed to the startup configuration and the switch will reboot automatically with the new priority queue changes in effect. See Table 6-14 on page 6-96 for a listing of the default GMB percentages that are allocated per queue.

Viewing the QoS Queue Configuration

To display the current priority queue configuration and memory allocations per queue, enter the **show qos queue-config** command.

```
ProCurve#: show qos queue-config

         802.1p
Queue    Priority    Memory %
-----    --------    --------
   1        1-2         10
   2        0,3         70
   3        4-5         10
   4        6-7         10
```

Figure 6-43. Displaying QoS Queue Configuration

Stack Management for the 3500, 3500yl, 6200yl and 6600 Switches

Contents

Introduction to Stack Management on the 3500, 3500yl, 6200yl and 6600 Switches 7-3
Components of ProCurve Stack Management 7-5
General Stacking Operation 7-5
Operating Rules for Stacking 7-7
General Rules 7-7
Specific Rules 7-8
Configuring Stack Management 7-9
Overview of Configuring and Bringing Up a Stack 7-9
General Steps for Creating a Stack 7-11
Using the Menu Interface To View Stack Status and Configure Stacking 7-13
Using the Menu Interface To View and Configure a Commander Switch 7-13
Using the Menu To Manage a Candidate Switch 7-15
Using the Commander To Manage The Stack 7-17
Using the Commander To Access Member Switches for Configuration Changes and Monitoring Traffic 7-23
Converting a Commander or Member to a Member of Another Stack 7-24
Monitoring Stack Status 7-25
Using the CLI To View Stack Status and Configure Stacking 7-29
Using the CLI To View Stack Status 7-31
Using the CLI To Configure a Commander Switch 7-33
Adding to a Stack or Moving Switches Between Stacks 7-35
Using the CLI To Remove a Member from a Stack 7-40

Using the CLI To Access Member Switches for Configuration Changes and Traffic Monitoring 7-42
SNMP Community Operation in a Stack 7-43
Using the CLI To Disable or Re-Enable Stacking 7-44
Transmission Interval 7-44
Stacking Operation with Multiple VLANs Configured 7-44
Status Messages 7-45

Introduction to Stack Management on the 3500, 3500yl, 6200yl and 6600 Switches

This feature is available on the 3500, 3500yl, 6200yl and 6600 switches, but not on the 5400zl and 8200zl switches.

ProCurve Stack Management (*stacking*) enables you to use a single IP address and standard network cabling to manage a group of up to 16 total switches in the same IP subnet (broadcast domain). Using stacking, you can:

- Reduce the number of IP addresses needed in your network.
- Simplify management of small workgroups or wiring closets while scaling your network to handle increased bandwidth demand.
- Eliminate any specialized cables for stacking connectivity and remove the distance barriers that typically limit your topology options when using other stacking technologies.
- Add switches to your network without having to first perform IP addressing tasks.

Notes

- Stacking and meshing cannot both be enabled at the same time.
- In the default configuration, stacking in the “candidate” state is enabled.
- For additional rules and restrictions, see “Operating Rules for Stacking” on page 7-7.

Summary of Stacking Features

Feature	Default	Menu	CLI	Web
view stack status				
view status of a single switch	n/a	page 7-26 thru page 7-28	page 7-31	Refer to Online Help
view candidate status	n/a		page 7-31	
view status of commander and its stack	n/a		page 7-32	
view status of all stacking-enabled switches in the ip subnet	n/a		page 7-32	
configure stacking				
enable/disable candidate Auto-Join	enabled/Yes	page 7-15	page 7-37	
"push" a candidate into a stack	n/a	page 7-15	page 7-37	
configure a switch to be a commander	n/a	page 7-13	page 7-33	
"push" a member into another stack	n/a	page 7-24	page 7-39	
remove a member from a stack	n/a	page 7-21	page 7-40 or page 7-41	
"pull" a candidate into a stack	n/a	page 7-17	page 7-36	
"pull" a member from another stack	n/a	page 7-19	page 7-38	
convert a commander or member to a member of another stack	n/a	page 7-24	page 7-39	
access member switches for configuration and traffic monitoring	n/a	page 7-23	page 7-42	
disable stacking	enabled	page 7-15	page 7-44	
transmission interval	60 seconds	page 7-13	page 7-44	

Components of ProCurve Stack Management

Table 7-1. Stacking Definitions

Stack	Consists of a Commander switch and any Member switches belonging to that Commander's stack.
Commander	A switch that has been manually configured as the controlling device for a stack. When this occurs, the switch's stacking configuration appears as **Commander**.
Candidate	A switch that is ready to join (become a Member of) a stack through either automatic or manual methods. A switch configured as a Candidate is not in a stack.
Member	A switch that has joined a stack and is accessible from the stack Commander.

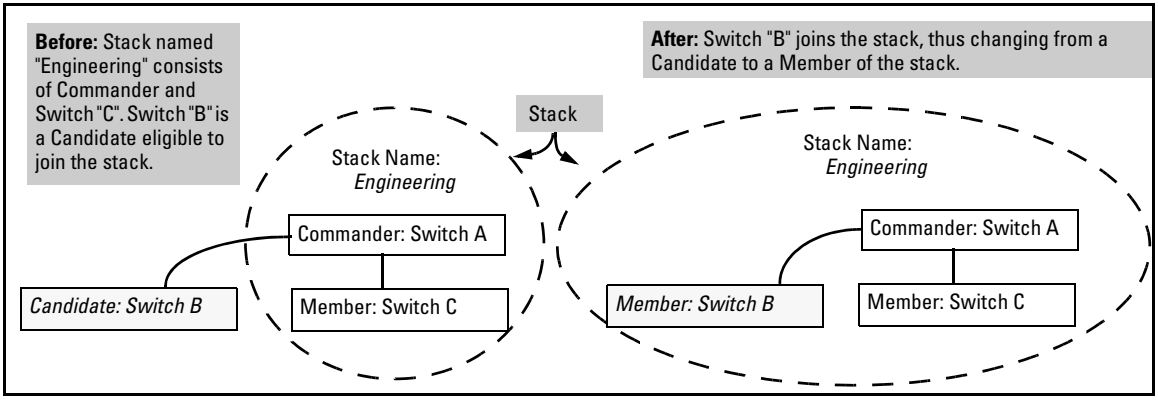

Before: Stack named "Engineering" consists of Commander and Switch "C". Switch "B" is a Candidate eligible to join the stack.
After: Switch "B" joins the stack, thus changing from a Candidate to a Member of the stack.
Stack
Stack Name: *Engineering*
Commander: Switch A
Member: Switch C
Candidate: Switch B
Stack Name: *Engineering*
Commander: Switch A
Member: Switch B
Member: Switch C

Figure 7-1. Illustration of a Switch Moving from Candidate to Member

General Stacking Operation

After you configure one switch to operate as the Commander of a stack, additional switches can join the stack by either automatic or manual methods. After a switch becomes a Member, you can work through the Commander switch to further configure the Member switch as necessary for all of the additional software features available in the switch.

The Commander switch serves as the in-band entry point for access to the Member switches. For example, the Commander's IP address becomes the path to all stack Members and the Commander's Manager password controls access to all stack Members.

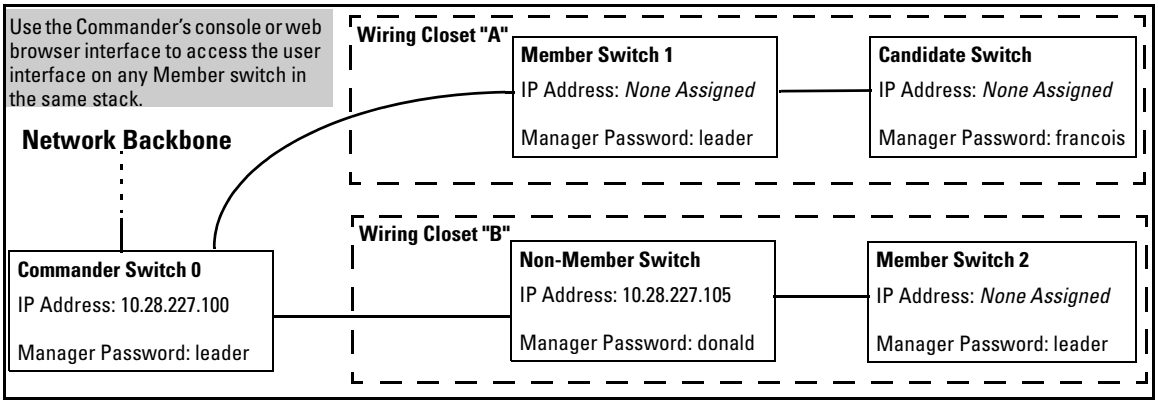


Figure 7-2. Example of Stacking with One Commander Controlling Access to Wiring Closet Switches

Interface Options. You can configure stacking through the switch's menu interface, CLI, or the web browser interface. For information on how to use the web browser interface to configure stacking, see the online Help for the web browser interface.

Web Browser Interface Window for Commander Switches. The web browser interface window for a Commander switch differs in appearance from the same window for non-commander switches.

Operating Rules for Stacking

General Rules

- Stacking is an optional feature (enabled in the default configuration) and can easily be disabled. Stacking has no effect on the normal operation of the switch in your network.
- A stack requires one Commander switch. (Only one Commander allowed per stack.)
- All switches in a particular stack must be in the same IP subnet (broadcast domain). A stack cannot cross a router.
- A stack accepts up to 16 switches (numbered 0-15), including the Commander (always numbered 0).
- The stacking feature supports up to 100 switches in the same IP subnet (broadcast domain), however, a switch can belong to only one stack. In the event that the 100 switch limit is exceeded, it may take multiple attempts to add or move a member to any given stack. Once a member is added to a stack, it is not "forgotten" by the Commander.
- The **stack status (all)** command will display up to 100 devices. Devices that are not members of a given stack may periodically "drop out" of the list.
- If multiple VLANs are configured, stacking uses only the primary VLAN on any switch. In the factory-default configuration, the DEFAULT_VLAN is the primary VLAN. (See "Stacking Operation with Multiple VLANs Configured" on page 7-44 and "The Primary VLAN" on page 2-51.)
- Stacking allows intermediate devices that do not support stacking. This enables you to include switches that are distant from the Commander.

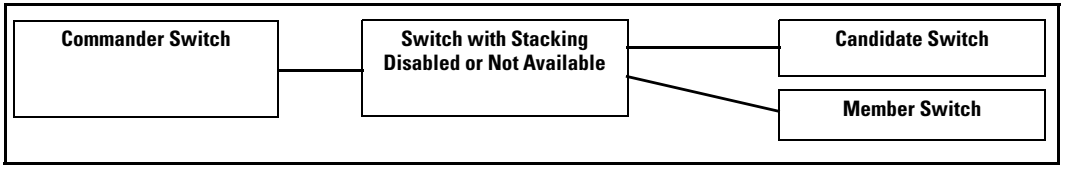


Figure 7-3. Example of a Non-Stacking Device Used in a Stacking Environment

Specific Rules

Table 7-2. Specific Rules for Commander, Candidate, and Member Switch

	IP Addressing and Stack Name	Number Allowed Per Stack	Passwords	SNMP Communities
Commander	**IP Addr:** Requires an assigned IP address and mask for access via the network. **Stack Name:** Required	Only one Commander switch is allowed per stack.	The Commander's Manager and Operator passwords are assigned to any switch becoming a Member of the stack. If you change the Commander's passwords, the Commander propagates the new passwords to all stack Members.	Standard SNMP community operation. The Commander also operates as an SNMP proxy to Members for all SNMP communities configured in the Commander.
Candidate	**IP Addr:** Optional. Configuring an IP address allows access via Telnet or web browser interface while the switch is not a stack member. In the factory default configuration the switch automatically acquires an IP address if your network includes DHCP service. **Stack Name:** N/A	n/a	Passwords optional. If the Candidate becomes a stack Member, it assumes the Commander's Manager and Operator passwords. If a candidate has a password, it cannot be automatically added to a stack. In this case, if you want the Candidate in a stack, you must manually add it to the stack.	Uses standard SNMP community operation if the Candidate has its own IP addressing.
Member	**IP Addr:** Optional. Configuring an IP address allows access via Telnet or web browser interface without going through the Commander switch. This is useful, for example, if the stack Commander fails and you need to convert a Member switch to operate as a replacement Commander. **Stack Name:** N/A	Up to 15 Members per stack.	When the switch joins the stack, it automatically assumes the Commander's Manager and Operator passwords and discards any passwords it may have had while a Candidate. **Note:** If a Member leaves a stack for any reason, it retains the passwords assigned to the stack Commander at the time of departure from the stack.	Belongs to the same SNMP communities as the Commander (which serves as an SNMP proxy to the Member for communities to which the Commander belongs). To join other communities that *exclude* the Commander, the Member must have its own IP address. Loss of stack membership means loss of membership in any community that is configured only in the Commander. See "SNMP Community Operation in a Stack" on page 7-43.

Note

In the default stack configuration, the Candidate **Auto Join** parameter is enabled, but the Commander **Auto Grab** parameter is disabled. This prevents Candidates from automatically joining a stack prematurely or joining the wrong stack (if more than one stack Commander is configured in a subnet or broadcast domain). If you plan to install more than one stack in a subnet, HP recommends that you leave **Auto Grab** disabled on all Commander switches and manually add Members to their stacks. Similarly, if you plan to install a stack in a subnet (broadcast domain) where stacking-capable switches are not intended for stack membership, you should set the **Stack State** parameter (in the Stack Configuration screen) to **Disabled** on those particular switches.

Configuring Stack Management

Overview of Configuring and Bringing Up a Stack

This process assumes that:

- All switches you want to include in a stack are connected to the same subnet (broadcast domain).
- If VLANs are enabled on the switches you want to include in the stack, then the ports linking the stacked switches must be on the primary VLAN in each switch (which, in the default configuration, is the default VLAN). If the primary VLAN is tagged, then each switch in the stack must use the same VLAN ID (VID) for the primary VLAN. (Refer to “The Primary VLAN” on page 2-51, and “Stacking Operation with Multiple VLANs Configured” on page 7-44.)
- *If you are including a ProCurve Switch 8000M, 4000M, 2424M, 2400M, or 1600M in a stack, you must first update all such devices to software version C.08.03 or later.* (You can get a copy of the latest software version from the ProCurve Networking web site and/or copy it from one switch to another. For downloading instructions, see appendix A, “File Transfers”, in the *Management and Configuration Guide* for your switch.)

Options for Configuring a Commander and Candidates. Depending on how Commander and Candidate switches are configured, Candidates can join a stack either automatically or by a Commander manually adding ("pulling") them into the stack. In the default configuration, a Candidate joins only when *manually* pulled by a Commander. You can reconfigure a Commander to *automatically* pull in Candidates that are in the default stacking configuration. You can also reconfigure a Candidate switch to either "push" itself into a particular Commander's stack, convert the Candidate to a Commander (for a stack that does not already have a Commander), or to operate as a standalone switch without stacking. The following table shows your control options for adding Members to a stack.

Table 7-3. Stacking Configuration Guide

Join Method[1]	Commander (IP Addressing Required)	Candidate (IP Addressing Optional)	
	Auto Grab	**Auto Join**	**Passwords**
Automatically add Candidate to Stack (Causes the first 15 eligible, discovered switches in the subnet to automatically join a stack.)	**Yes**	**Yes** *(default)*	No *(default)*[*]
Manually add Candidate to Stack (Prevent automatic joining of switches you don't want in the stack)	**No** *(default)*	**Yes** *(default)*	Optional[*]
	Yes	**No**	Optional[*]
	Yes	**Yes** *(default)* or **No**	Configured
Prevent a switch from being a Candidate	**N/A**	**Disabled**	Optional

[*]The Commander's Manager and Operator passwords propagate to the candidate when it joins the stack.

The easiest way to *automatically* create a stack is to:

1. Configure a switch as a Commander.
2. Configure IP addressing and a stack name on the Commander.
3. Set the Commander's **Auto Grab** parameter to **Yes**.
4. Connect Candidate switches (in their factory default configuration) to the network.

This approach automatically creates a stack of up to 16 switches (including the Commander). However this replaces manual control with an automatic process that may bring switches into the stack that you did not intend to include. With the Commander's **Auto Grab** parameter set to **Yes**, *any switch* conforming to all four of the following factors automatically becomes a stack Member:

- Default stacking configuration (**Stack State** set to **Candidate**, and **Auto Join** set to **Yes**)
- Same subnet (broadcast domain) and default VLAN as the Commander (If VLANs are used in the stack environment, see "Stacking Operation with a Tagged VLAN" on page 7-44.)
- No Manager password
- 14 or fewer stack members at the moment

General Steps for Creating a Stack

This section describes the general stack creation process. For the detailed configuration processes, see pages 7-13 through 7-36 for the menu interface and pages 7-29 through 7-41 for the CLI.

1. Determine the naming conventions for the stack. You will need a stack name. Also, to help distinguish one switch from another in the stack, you can configure a unique system name for each switch. Otherwise, the system name for a switch appearing in the Stacking Status screen appears as the stack name plus an automatically assigned switch number. For example:

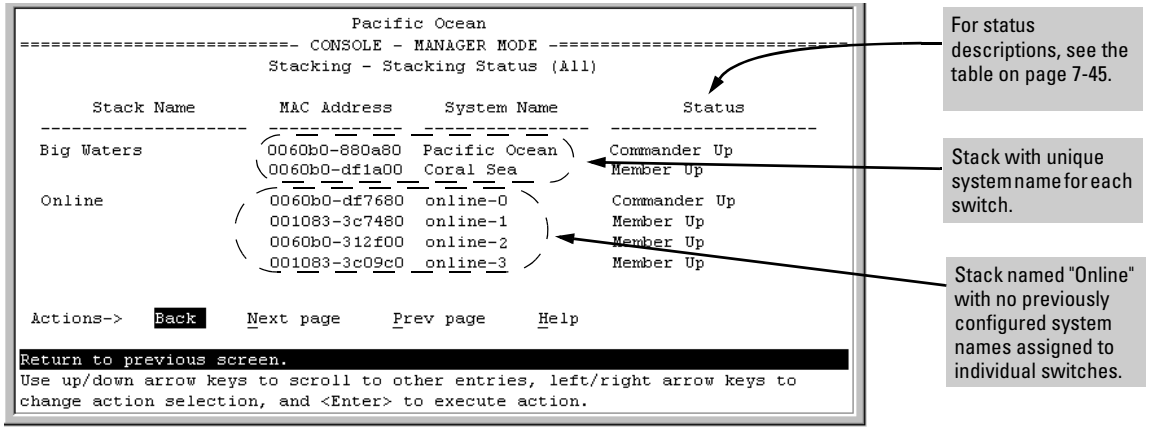


Figure 7-4. Using the System Name to Help Identify Individual Switches

2. Configure the Commander switch. Doing this first helps to establish consistency in your stack configuration, which can help prevent startup problems.
   - A stack requires one Commander switch. If you plan to implement more than one stack in a subnet (broadcast domain), the easiest way to avoid unintentionally adding a Candidate to the wrong stack is to manually control the joining process by leaving the Commander's **Auto Grab** parameter set to **No** (the default).
   - The Commander assigns its Manager and Operator passwords to any Candidate switch that joins the stack.
   - The Commander's SNMP community names apply to members.
3. For automatically or manually pulling Candidate switches into a stack, you can leave such switches in their default stacking configuration. If you need to access Candidate switches through your network before they join the stack, assign IP addresses to these devices. Otherwise, IP addressing is optional for Candidates and Members. (Note that once a Candidate becomes a member, you can access it through the Commander to assign IP addressing or make other configuration changes.)
4. Make a record of any Manager passwords assigned to the switches (intended for your stack) that are not currently members. (You will use these passwords to enable the protected switches to join the stack.)
5. If you are using VLANs in the stacking environment, you must use the default VLAN for stacking links. For more information, see "Stacking Operation with a Tagged VLAN" on page 7-44.
6. Ensure that all switches intended for the stack are connected to the same subnet (broadcast domain). As soon as you connect the Commander, it will begin discovering the available Candidates in the subnet.
   - If you configured the Commander to automatically add Members (**Auto Grab** = **Yes**), the first fifteen discovered Candidates meeting both of the following criteria will automatically join the stack:
     - **Auto Join** parameter set to **Yes** (the default)
     - Manager password not configured
   - If you configured the Commander to manually add Members (**Auto Grab** set to **No**—the default), you can begin the process of selecting and adding the desired Candidates.
7. Ensure that all switches intended for the stack have joined.
8. If you need to do specific configuration or monitoring tasks on a Member, use the console interface on the Commander to access the Member.

Using the Menu Interface To View Stack Status and Configure Stacking

Using the Menu Interface To View and Configure a Commander Switch

1. Configure an IP address and subnet mask on the Commander switch. (Refer to the *Management and Configuration Guide* for your switch.)
2. Display the Stacking Menu by selecting **Stacking** in the Main Menu.

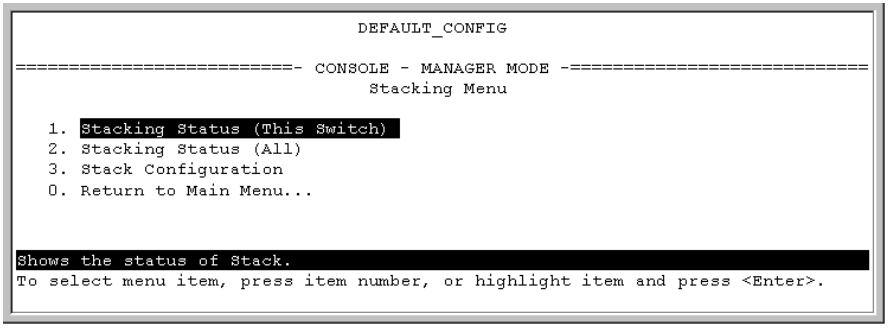

DEFAULT_CONFIG
CONSOLE - MANAGER MODE
Stacking Menu
1. Stacking Status (This Switch)
2. Stacking Status (All)
3. Stack Configuration
0. Return to Main Menu...
Shows the status of Stack.
To select menu item, press item number, or highlight item and press <Enter>.

Figure 7-5. The Default Stacking Menu

3. Display the Stack Configuration menu by pressing **[3]** to select **Stack Configuration**.

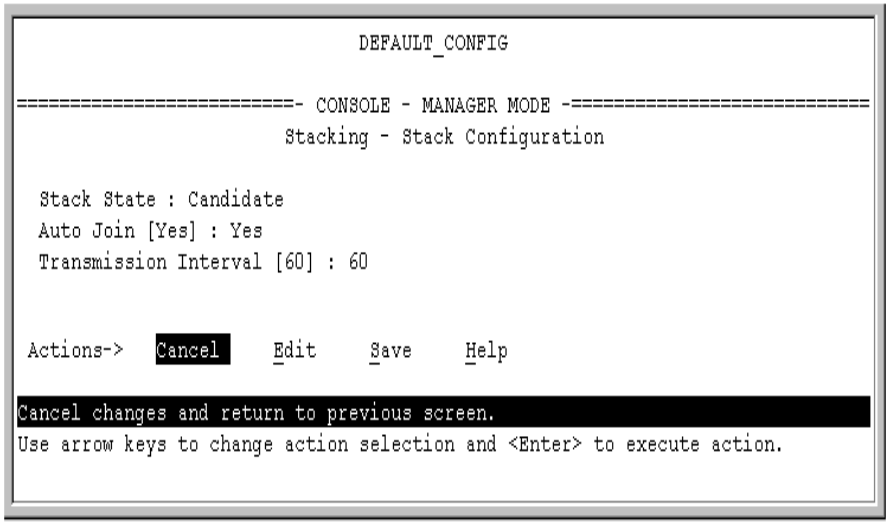

DEFAULT_CONFIG
CONSOLE - MANAGER MODE
Stacking - Stack Configuration
Stack State : Candidate
Auto Join [Yes] : Yes
Transmission Interval [60] : 60
Actions-> Cancel Edit Save Help
Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.

Figure 7-6. The Default Stack Configuration Screen

4. Move the cursor to the Stack State field by pressing **[E]** (for **Edit**). Then use the Space bar to select the **Commander** option.
5. Press the downarrow key to display the Commander configuration fields in the Stack Configuration screen.

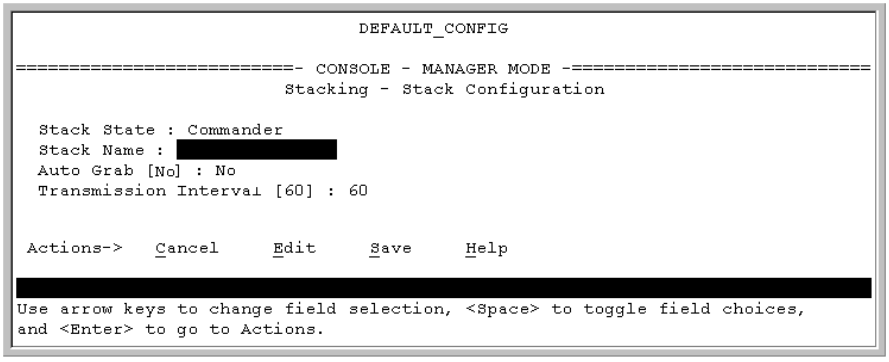

DEFAULT_CONFIG
CONSOLE - MANAGER MODE
Stacking - Stack Configuration
Stack State : Commander
Stack Name :
Auto Grab [No] : No
Transmission Interval [60] : 60
Actions-> Cancel Edit Save Help
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.

Figure 7-7. The Default Commander Configuration in the Stack Configuration Screen

6. Enter a unique stack name (up to 15 characters; no spaces) and press the downarrow key.
7. Ensure that the Commander has the desired **Auto Grab** setting, then press the downarrow key:
   - **No** (the default) prevents automatic joining of Candidates that have their **Auto Join** set to **Yes**.
   - **Yes** enables the Commander to automatically take a Candidate into the stack as a Member if the Candidate has **Auto Join** set to **Yes** (the default Candidate setting) and does not have a previously configured password.
8. Accept or change the transmission interval (default: 60 seconds), then press **[Enter]** to return the cursor to the **Actions** line.
9. Press **[S]** (for **Save**) to save your configuration changes and return to the Stacking menu.

Your Commander switch should now be ready to automatically or manually acquire Member switches from the list of discovered Candidates, depending on your configuration choices.

Using the Menu To Manage a Candidate Switch

Using the menu interface, you can perform these actions on a Candidate switch:

- Add ("push") the Candidate into an existing stack
- Modify the Candidate's stacking configuration (**Auto Join** and **Transmission Interval**)
- Convert the Candidate to a Commander
- Disable stacking on the Candidate so that it operates as a standalone switch

In its default stacking configuration, a Candidate switch can either automatically join a stack or be manually added ("pulled") into a stack by a Commander, depending on the Commander's **Auto Grab** setting. The following table lists the Candidate's configuration options:

Table 7-4. Candidate Configuration Options in the Menu Interface

Parameter	Default Setting	Other Settings
Stack State	Candidate	Commander, Member, or Disabled
Auto Join	Yes	No
Transmission Interval	60 Seconds	Range: 1 to 300 seconds

Using the Menu To "Push" a Switch Into a Stack, Modify the Switch's Configuration, or Disable Stacking on the Switch. Use Telnet or the web browser interface to access the Candidate if it has an IP address. Otherwise, use a direct connection from a terminal device to the switch's console port. (For information on how to use the web browser interface, see the online Help provided for the browser.)

1. Display the Stacking Menu by selecting **Stacking** in the console Main Menu.
2. Display the Stack Configuration menu by pressing **[3]** to select **Stack Configuration**.

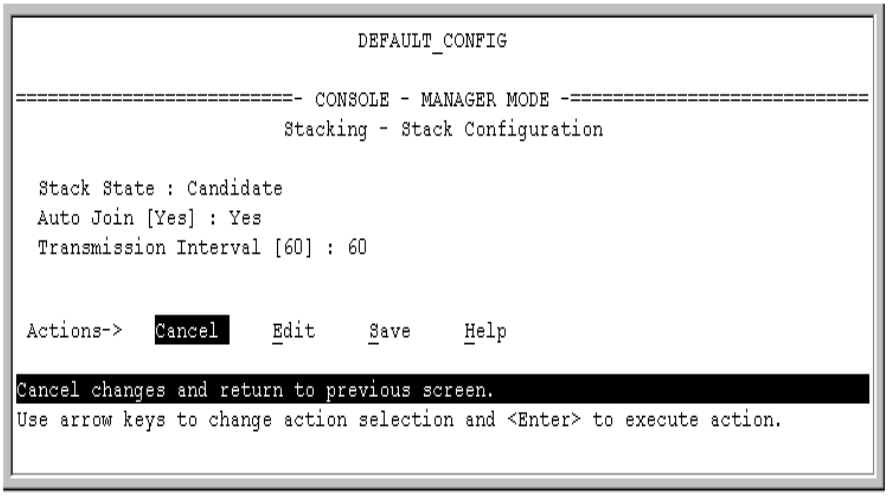

DEFAULT_CONFIG
CONSOLE - MANAGER MODE
Stacking - Stack Configuration
Stack State : Candidate
Auto Join [Yes] : Yes
Transmission Interval [60] : 60
Actions-> Cancel Edit Save Help
Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.

Figure 7-8. The Default Stack Configuration Screen

3. Move the cursor to the Stack State field by pressing **[E]** (for **Edit**).
4. Do one of the following:
   - To disable stacking on the Candidate, use the Space bar to select the **Disabled** option, then go to step 5.

     Note: Using the menu interface to disable stacking on a Candidate removes the Candidate from all stacking menus.
   - To insert the Candidate into a specific Commander's stack:
     i. Use the space bar to select Member.
     ii. Press **[Tab]** once to display the **Commander MAC Address** parameter, then enter the MAC address of the desired Commander.
   - To change **Auto Join** or **Transmission Interval**, use **[Tab]** to select the desired parameter, and:
     – To change **Auto Join**, use the Space bar.
     – To change **Transmission Interval**, type in the new value in the range of 1 to 300 seconds.

       Note: All switches in the stack must be set to the same transmission interval to help ensure proper stacking operation. HP recommends that you leave this parameter set to the default 60 seconds.

   Then go to step 5.
5. press **[Enter]** to return the cursor to the **Actions** line.

6. Press **[S]** (for **Save**) to save your configuration changes and return to the Stacking menu.

Using the Commander To Manage The Stack

The Commander normally operates as your stack manager and point of entry into other switches in the stack. This typically includes:

- Adding new stack members
- Moving members between stacks
- Removing members from a stack
- Accessing stack members for individual configuration changes and traffic monitoring

The Commander also imposes its passwords on all stack members and provides SNMP community membership to the stack. (See "SNMP Community Operation in a Stack" on page 7-43.)

Using the Commander's Menu To Manually Add a Candidate to a Stack. In the default configuration, you must manually add stack Members from the Candidate pool. Reasons for a switch remaining a Candidate instead of becoming a Member include any of the following:

- **Auto Grab** in the Commander is set to **No** (the default).
- **Auto Join** in the Candidate is set to **No**.

  Note: When a switch leaves a stack and returns to Candidate status, its **Auto Join** parameter resets to **No** so that it will not immediately rejoin a stack from which it has just departed.
- A Manager password is set in the Candidate.
- The stack is full.

Unless the stack is already full, you can use the Stack Management screen to manually convert a Candidate to a Member. If the Candidate has a Manager password, you will need to use it to make the Candidate a Member of the stack.

1. To add a Member, start at the Main Menu and select:

   9. Stacking...

   4. Stack Management

   You will then see the Stack Management screen:

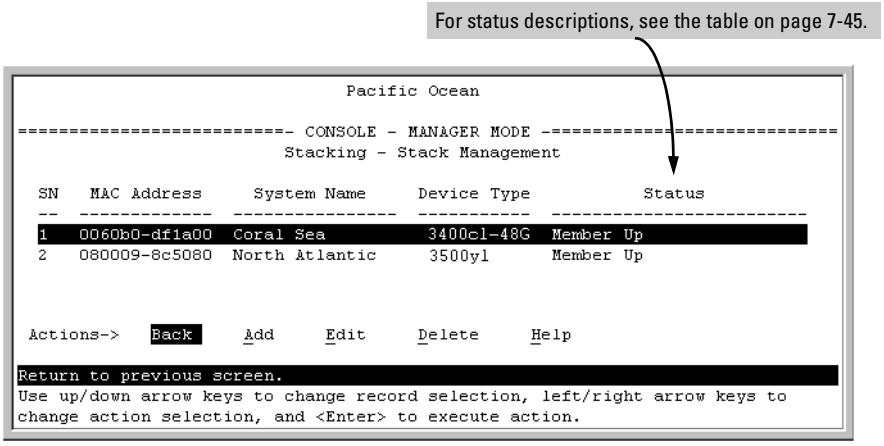

For status descriptions, see the table on page 7-45.
Pacific Ocean
CONSOLE - MANAGER MODE
Stacking - Stack Management
SN MAC Address System Name Device Type Status
1 0060b0-df1a00 Coral Sea 3400cl-48G Member Up
2 080009-8c5080 North Atlantic 3500yl Member Up
Actions-> Back Add Edit Delete Help
Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 7-9. Example of the Stack Management Screen

2. Press **[A]** (for **Add**) to add a Candidate. You will then see this screen listing the available Candidates:

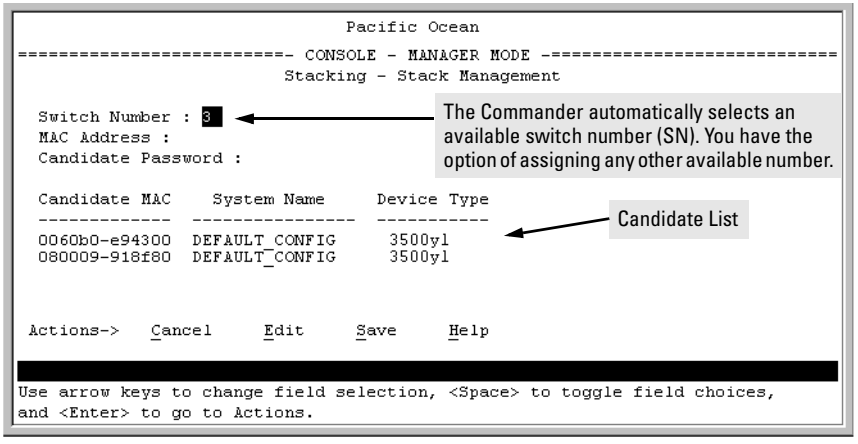

Pacific Ocean
CONSOLE - MANAGER MODE
Stacking - Stack Management
Switch Number : 3
MAC Address :
Candidate Password :
The Commander automatically selects an available switch number (SN). You have the option of assigning any other available number.
Candidate MAC System Name Device Type
0060b0-e94300 DEFAULT_CONFIG 3500yl
080009-918f80 DEFAULT_CONFIG 3500yl
Candidate List
Actions-> Cancel Edit Save Help
Use arrow keys to change field selection, <Space> to toggle field choices, and <Enter> to go to Actions.

Figure 7-10. Example of Candidate List in Stack Management Screen

3. Either accept the displayed switch number or enter another available number. (The range is 0 - 15, with 0 reserved for the Commander.)
4. Use the downarrow key to move the cursor to the MAC Address field, then type the MAC address of the desired Candidate from the Candidate list in the lower part of the screen.
5. Do one of the following:

- If the desired Candidate has a Manager password, press the downarrow key to move the cursor to the Candidate Password field, then type the password.
- If the desired Candidate does not have a password, go to step 6.

6. Press **[Enter]** to return to the Actions line, then press **[S]** (for **Save**) to complete the Add process for the selected Candidate. You will then see a screen similar to the one in figure 7-11, below, with the newly added Member listed.

   Note: If the message **Unable to add stack member: Invalid Password** appears in the console menu's Help line, then you either omitted the Candidate's Manager password or incorrectly entered the Manager password.

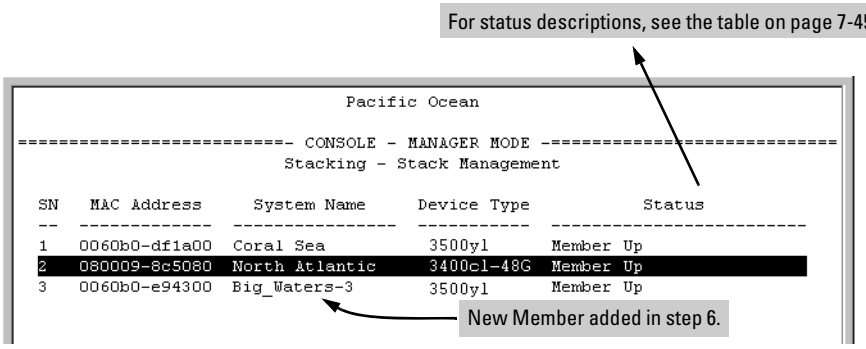

For status descriptions, see the table on page 7-45.
Pacific Ocean
CONSOLE - MANAGER MODE
Stacking - Stack Management
SN MAC Address System Name Device Type Status
1 0060b0-df1a00 Coral Sea 3500yl Member Up
2 080009-8c5080 North Atlantic 3400cl-48G Member Up
3 0060b0-e94300 Big_Waters-3 3500yl Member Up
New Member added in step 6.

Figure 7-11. Example of Stack Management Screen After New Member Added

Using the Commander's Menu To Move a Member From One Stack to Another. Where two or more stacks exist in the same subnet (broadcast domain), you can easily move a Member of one stack to another stack if the destination stack is not full. (If you are using VLANs in your stack environment, see "Stacking Operation with a Tagged VLAN" on page 7-44.) This procedure is nearly identical to manually adding a Candidate to a stack (page 7-17). (If the stack from which you want to move the Member has a Manager password, you will need to know the password to make the move.)

1. To move a Member from one stack to another, go to the Main Menu of the Commander in the destination stack and display the Stacking Menu by selecting

   9. Stacking...

2. To learn or verify the MAC address of the Member you want to move, display a listing of all Commanders, Members, and Candidates in the subnet by selecting:

2. Stacking Status (All)

You will then see the Stacking Status (All) screen:

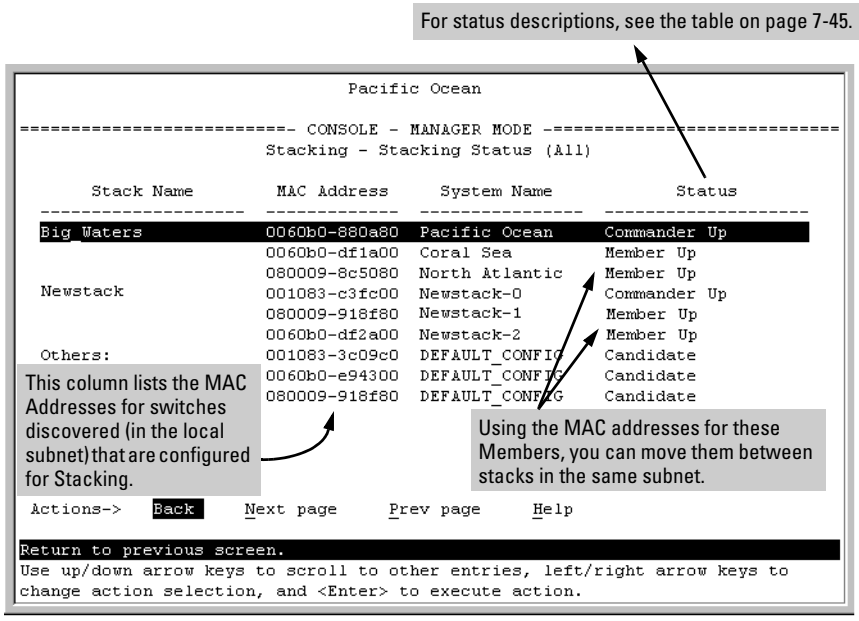


Figure 7-12. Example of How the Stacking Status (All) Screen Helps You Find Member MAC Addresses

3. In the Stacking Status (All) screen, find the Member switch that you want to move and note its MAC address, then press **[B]** (for **Back**) to return to the Stacking Menu.
4. Display the Commander's Stack Management screen by selecting

   4. Stack Management

   (For an example of this screen, see figure 7-9 on page 7-18.)
5. Press **[A]** (for **Add**) to add the Member. You will then see a screen listing any available candidates. (See figure 7-10 on page 7-18.) Note that you will not see the switch you want to add because it is a Member of another stack and not a Candidate.)
6. Either accept the displayed switch number or enter another available number. (The range is 0 - 15, with 0 reserved for the Commander.)
7. Use the downarrow key to move the cursor to the MAC Address field, then type the MAC address of the desired Member you want to move from another stack.

8. Do one of the following:
   - If the stack containing the Member you are moving has a Manager password, press the downarrow key to select the Candidate Password field, then type the password.
   - If the stack containing the Member you want to move does not have a password, go to step 9.
9. Press **[Enter]** to return to the Actions line, then press **[S]** (for **Save**) to complete the Add process for the selected Member. You will then see a screen similar to the one in figure 7-9 on page 7-18, with the newly added Member listed.

Note: If the message **Unable to add stack member: Invalid Password** appears in the console menu's Help line, then you either omitted the Manager password for the stack containing the Member or incorrectly entered the Manager password.

You can "push" a Member from one stack to another by going to the Member's interface and entering the MAC address of the destination stack Commander in the Member's **Commander MAC Address** field. Using this method moves the Member to another stack without a need for knowing the Manager password in that stack, but also blocks access to the Member from the original Commander.

Using the Commander's Menu To Remove a Stack Member. These rules affect removals from a stack:

- When a Candidate becomes a Member, its **Auto Join** parameter is automatically set to **No**. This prevents the switch from automatically rejoining a stack as soon as you remove it from the stack.
- When you use the Commander to remove a switch from a stack, the switch rejoins the Candidate pool for your IP subnet (broadcast domain), with **Auto Join** set to **No**.
- When you remove a Member from a stack, it frees the previously assigned switch number (**SN**), which then becomes available for assignment to another switch that you may subsequently add to the stack. The default switch number used for an add is the lowest unassigned number in the Member range (1 - 15; 0 is reserved for the Commander).

To remove a Member from a stack, use the Stack Management screen.

1. From the Main Menu, select:

   9. Stacking...

   4. Stack Management

   You will then see the Stack Management screen:

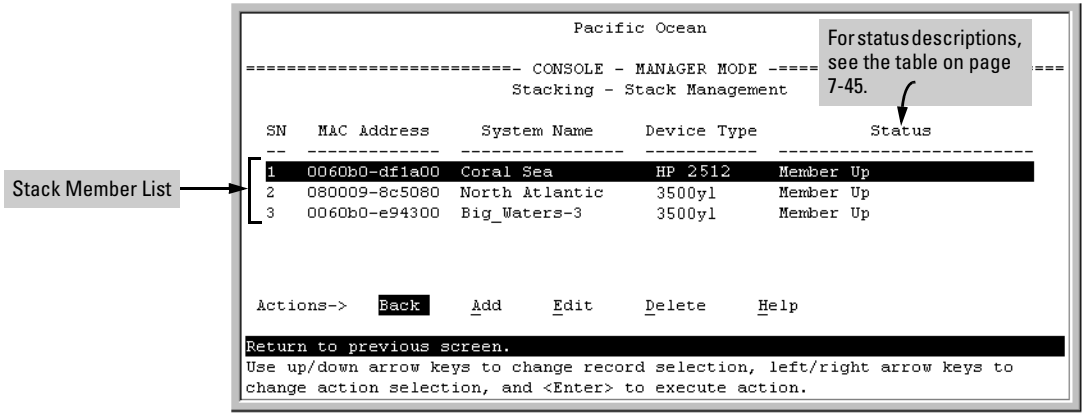

Pacific Ocean
For status descriptions, see the table on page 7-45.
CONSOLE - MANAGER MODE
Stacking - Stack Management
SN MAC Address System Name Device Type Status
1 0060b0-df1a00 Coral Sea HP 2512 Member Up
2 080009-8c5080 North Atlantic 3500yl Member Up
3 0060b0-e94300 Big_Waters-3 3500yl Member Up
Stack Member List
Actions-> Back Add Edit Delete Help
Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 7-13. Example of Stack Management Screen with Stack Members Listed

2. Use the downarrow key to select the Member you want to remove from the stack.

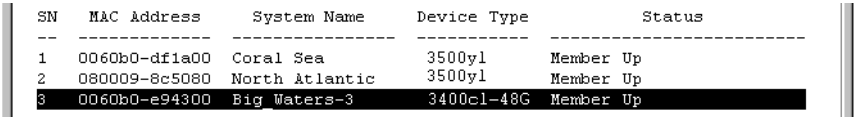

SN MAC Address System Name Device Type Status
1 0060b0-df1a00 Coral Sea 3500yl Member Up
2 080009-8c5080 North Atlantic 3500yl Member Up
3 0060b0-e94300 Big_Waters-3 3400cl-48G Member Up

Figure 7-14. Example of Selecting a Member for Removal from the Stack

3. Type **[D]** (for **Delete**) to remove the selected Member from the stack. You will then see the following prompt:

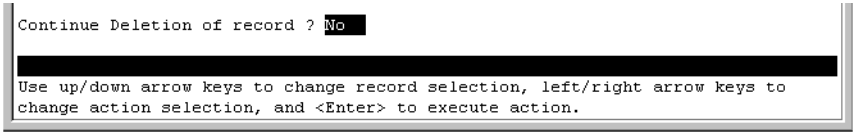

Continue Deletion of record ? No
Use up/down arrow keys to change record selection, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 7-15. The Prompt for Completing the Deletion of a Member from the Stack

4. To continue deleting the selected Member, press the Space bar once to select **Yes** for the prompt, then press **[Enter]** to complete the deletion. The Stack Management screen updates to show the new stack Member list.

Using the Commander To Access Member Switches for Configuration Changes and Monitoring Traffic

After a Candidate becomes a stack Member, you can use that stack's Commander to access the Member's console interface for the same configuration and monitoring that you would do through a Telnet or direct-connect access.

1. From the Main Menu, select:

   9. Stacking...
   5. Stack Access

   You will then see the Stack Access screen:

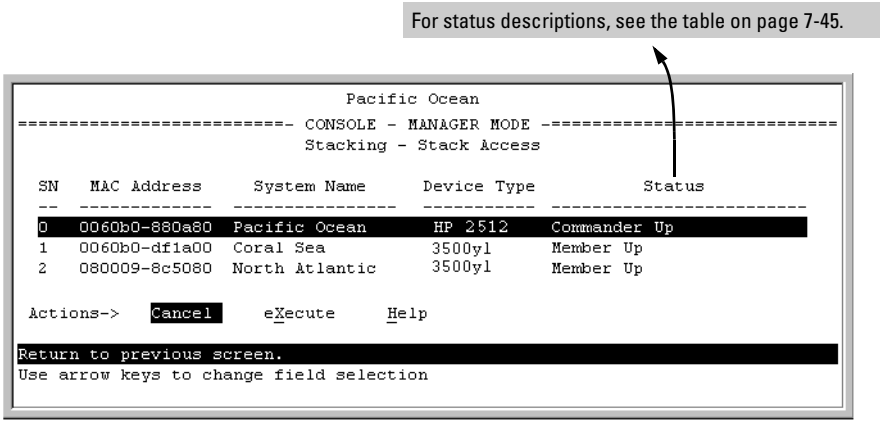

For status descriptions, see the table on page 7-45.
Pacific Ocean
CONSOLE - MANAGER MODE
Stacking - Stack Access
SN MAC Address System Name Device Type Status
0 0060b0-880a80 Pacific Ocean HP 2512 Commander Up
1 0060b0-df1a00 Coral Sea 3500yl Member Up
2 080009-8c5080 North Atlantic 3500yl Member Up
Actions-> Cancel eXecute Help
Return to previous screen.
Use arrow keys to change field selection

Figure 7-16. Example of the Stack Access Screen

Use the down arrow key to select the stack Member you want to access, then press **[X]** (for **eXecute**) to display the console interface for the selected Member. For example, if you selected switch number 1 (system name: **Coral Sea**) in figure 7-16 and then pressed **[X]**, you would see the Main Menu for the switch named Coral Sea.

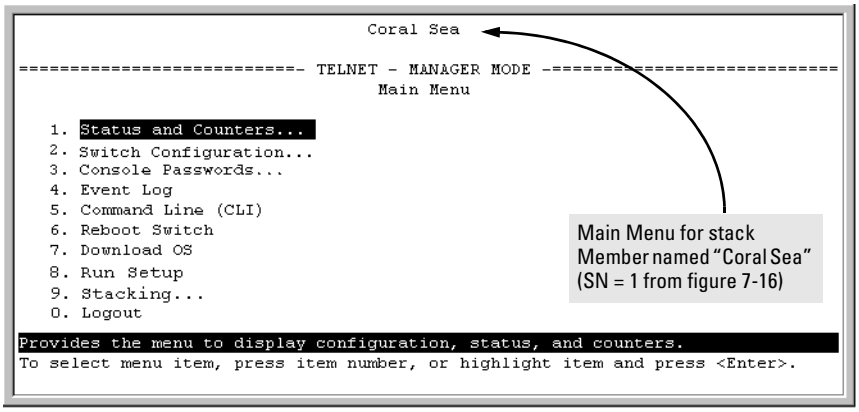

Coral Sea
TELNET - MANAGER MODE
Main Menu
1. Status and Counters...
2. Switch Configuration...
3. Console Passwords...
4. Event Log
5. Command Line (CLI)
6. Reboot Switch
7. Download OS
8. Run Setup
9. Stacking...
0. Logout
Provides the menu to display configuration, status, and counters.
To select menu item, press item number, or highlight item and press <Enter>.
Main Menu for stack Member named "Coral Sea" (SN = 1 from figure 7-16)

Figure 7-17. The eXecute Command Displays the Console Main Menu for the Selected Stack Member

2. You can now make configuration changes and/or view status data for the selected Member in the same way that you would if you were directly connected or telnetted into the switch.

3. When you are finished accessing the selected Member, do the following to return to the Commander's Stack Access screen:

   a. Return to the Member's Main Menu.

   b. Press **[0]** (for Logout), then **[Y]** (for Yes).

   c. Press **[Return]**.

   You should now see the Commander's Stack Access screen. (For an example, see figure 7-16 on page 7-23.)

Converting a Commander or Member to a Member of Another Stack

When moving a commander, the following procedure returns the stack members to Candidate status (with Auto-Join set to "**No**") and converts the stack Commander to a Member of another stack. When moving a member, the procedure simply pulls a Member out of one stack and pushes it into another.

1. From the Main Menu of the switch you want to move, select

   9. Stacking

2. To determine the MAC address of the destination Commander, select

   2. Stacking Status (All)

3. Press **[B]** (for **Back**) to return to the Stacking Menu.
4. To display Stack Configuration menu for the switch you are moving, select

   3. Stack Configuration

5. Press **[E]** (for **Edit**) to select the Stack State parameter.
6. Use the Space bar to select **Member**, then press ↓ to move to the **Commander MAC Address** field.
7. Enter the MAC address of the destination Commander and press **[Enter]**.
8. Press **[S]** (for **Save**).

Monitoring Stack Status

Using the stacking options in the menu interface for any switch in a stack, you can view stacking data for that switch or for all stacks in the subnet (broadcast domain). (If you are using VLANs in your stack environment, see "Stacking Operation with a Tagged VLAN" on page 7-44.) This can help you in such ways as determining the stacking configuration for individual switches, identifying stack Members and Candidates, and determining the status of individual switches in a stack. See table 7-5 on page 7-25.

Table 7-5. Stack Status Environments

Screen Name	Commander	Member	Candidate
Stack Status (This Switch)	• Commander's stacking configuration • Data on stack Members: – Switch Number – MAC Address – System Name – Device Type – Status	• Member's stacking configuration • Member Status • Data identifying Member's Commander: – Commander Status – Commander IP Address – Commander MAC Address	Candidate's stacking configuration
Stack Status (All)	Lists devices by stack name or Candidate status (if device is not a stack Member). Includes: • Stack Name • MAC Address • System Name • Status	Same as for Commander.	Same as for Commander.

Using Any Stacked Switch To View the Status for All Switches with Stacking Enabled. This procedure displays the general status of all switches in the IP subnet (broadcast domain) that have stacking enabled.

1. Go to the console Main Menu for any switch configured for stacking and select:

   9. Stacking ...

   2. Stacking Status (All)

   You will then see a Stacking Status screen similar to the following:

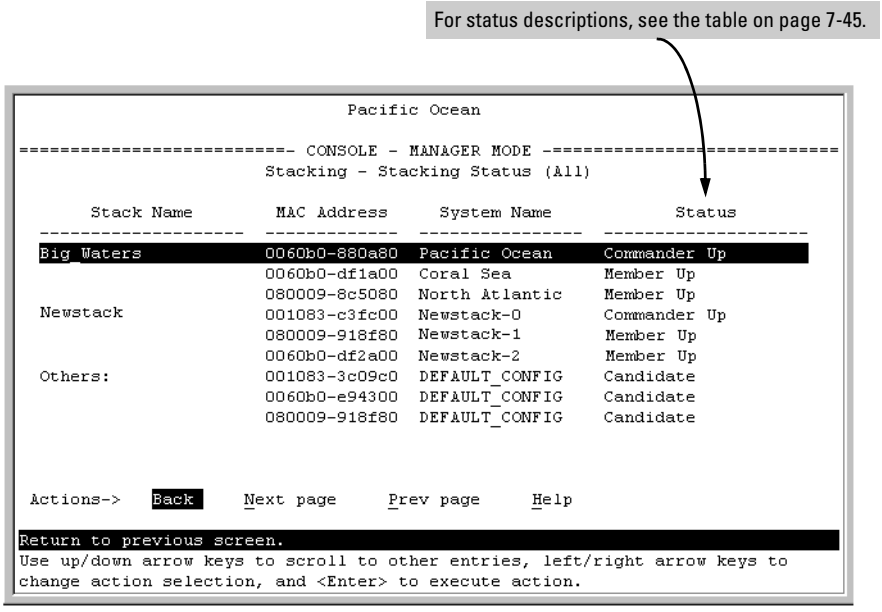

For status descriptions, see the table on page 7-45.
Pacific Ocean
CONSOLE - MANAGER MODE
Stacking - Stacking Status (All)
Stack Name
MAC Address
System Name
Status
Big_Waters 0060b0-880a80 Pacific Ocean Commander Up
0060b0-df1a00 Coral Sea Member Up
080009-8c5080 North Atlantic Member Up
Newstack 001083-c3fc00 Newstack-0 Commander Up
080009-918f80 Newstack-1 Member Up
0060b0-df2a00 Newstack-2 Member Up
Others: 001083-3c09c0 DEFAULT_CONFIG Candidate
0060b0-e94300 DEFAULT_CONFIG Candidate
080009-918f80 DEFAULT_CONFIG Candidate
Actions-> Back Next page Prev page Help
Return to previous screen.
Use up/down arrow keys to scroll to other entries, left/right arrow keys to change action selection, and <Enter> to execute action.

Figure 7-18. Example of Stacking Status for All Detected Switches Configured for Stacking

Viewing Commander Status. This procedure displays the Commander and stack configuration, plus information identifying each stack member.

To display the status for a Commander, go to the console Main Menu for the switch and select:

9. Stacking ...

1. Stacking Status (This Switch)

You will then see the Commander's Stacking Status screen:

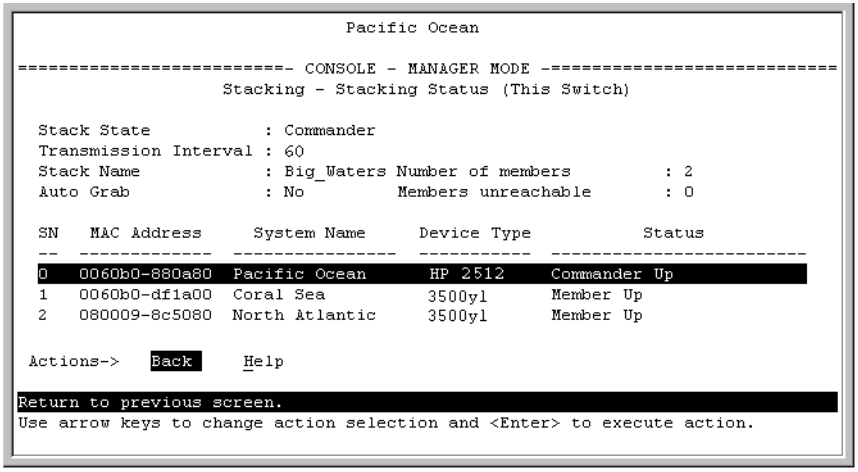

Pacific Ocean
CONSOLE - MANAGER MODE
Stacking - Stacking Status (This Switch)
Stack State : Commander
Transmission Interval : 60
Stack Name : Big_Waters Number of members : 2
Auto Grab : No Members unreachable : 0
SN MAC Address System Name Device Type Status
0 0060b0-880a80 Pacific Ocean HP 2512 Commander Up
1 0060b0-df1a00 Coral Sea 3500yl Member Up
2 080009-8c5080 North Atlantic 3500yl Member Up
Actions-> Back Help
Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.

Figure 7-19. Example of the Commander's Stacking Status Screen

Viewing Member Status. This procedure displays the Member's stacking information plus the Commander's status, IP address, and MAC address.

To display the status for a Member:

1. Go to the console Main Menu of the Commander switch and select

   9. Stacking ...

   5. Stack Access

2. Use the downarrow key to select the Member switch whose status you want to view, then press **[X]** (for **eXecute**). You will then see the Main Menu for the selected Member switch.

3. In the Member's Main Menu screen, select

   9. Stacking ...

   1. Stacking Status (This Switch)

   You will then see the Member's Stacking Status screen:

```
                                Coral Sea

============================= TELNET - MANAGER MODE ==============================
                     Stacking - Stacking Status (This Switch)

  Stack State                  : Member
  Transmission Interval        : 60
  Switch Number                : 1
  Stack Name                   : Big_Waters
  Member Status                : Joined Successfully
  Commander Status             : Commander Up
  Commander IP Address         : 10.28.227.102
  Commander MAC Address        : 0060b0-880a80


 Actions->   Back     Help

Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 7-20. Example of a Member's Stacking Status Screen

Viewing Candidate Status. This procedure displays the Candidate's stacking configuration.

To display the status for a Candidate:

1. Use Telnet (if the Candidate has a valid IP address for your network) or a direct serial port connection to access the menu interface Main Menu for the Candidate switch and select

   9. Stacking ...

   1. Stacking Status (This Switch)

   You will then see the Candidate's Stacking Status screen:

```
                                Coral Sea

============================= TELNET - MANAGER MODE ==============================
                     Stacking - Stacking Status (This Switch)

  Stack State           : Candidate
  Transmission Interval : 60
  Auto Join             : No


 Actions->   Back     Help

Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 7-21. Example of a Candidate's Stacking Screen

Using the CLI To View Stack Status and Configure Stacking

The CLI enables you to do all of the stacking tasks available through the menu interface.)

Table 7-6. CLI Commands for Configuring Stacking on a Switch

CLI Command	Operation
show stack [candidates | view | all]	**Commander:** Shows Commander's stacking configuration and lists the stack members and their individual status. **Member:** Lists Member's stacking configuration and status, and the status and the IP address and subnet mask of the stack Commander. Options: **candidates:** (Commander only) Lists stack Candidates. **view:** (Commander only) Lists current stack Members and their individual status. **all:** Lists all stack Commanders, Members and Candidates, with their individual status.
[no] stack	**Any Stacking-Capable Switch:** Enables or disables stacking on the switch. **Default:** Stacking Enabled
[no] stack commander *<stack name>*	**Candidate or Commander:** Converts a Candidate to a Commander or changes the stack name of an existing commander. "**No**" form eliminates named stack and returns Commander and stack Members to Candidate status with **Auto Join** set to **No**. "**No**" form prevents the switch from being discovered as a stacking-capable switch. **Default:** Switch Configured as a Candidate
[no] stack auto-grab	**Commander:** Causes Commander to automatically add to its stack any discovered Candidate in the subnet that does not have a Manager password and has **Auto-Join** set to **Yes**. **Default:** Disabled **Note:** If the Commander's stack already has 15 members, the Candidate cannot join until an existing member leaves the stack.

CLI Command	Operation
[no] stack member *<switch-num>* mac-address *<mac-addr>* [password *<password-str>*]	**Commander:** Adds a Candidate to stack membership. "No" form removes a Member from stack membership. To easily determine the MAC address of a Candidate, use the **show stack candidates** command. To determine the MAC address of a Member you want to remove, use the **show stack view** command.The password (*password-str*) is required only when adding a Candidate that has a Manager password.
telnet *<1..15>* *Used In:* Commander Only	**Commander:** Uses the **SN** (switch number— assigned by the stack Commander) to access the console interface (menu interface or CLI) of a stack member. To view the list of **SN** assignments for a stack, execute the **show stack** command in the Commander's CLI.
[no] stack join *<mac-addr>*	**Candidate:** Causes the Candidate to join the stack whose Commander has the indicated MAC address. "No" form is used in a Member to remove it from the stack of the Commander having the specified address. **Member:** "Pushes" the member to another stack whose Commander has the indicated MAC address.
[no] stack auto-join	**Candidate:** Enables Candidate to automatically join the stack of any Commander in the IP subnet that has **Auto Grab** enabled, or disables **Auto-Join** in the candidate. **Default: Auto Join** enabled. **Note:** If the Candidate has a Manager password or if the available stack(s) already have the maximum of 15 Members, the automatic join will not occur.
stack transmission-interval	**All Stack Members:** specifies the interval in seconds for transmitting stacking discovery packets. **Default:** 60 seconds

Using the CLI To View Stack Status

You can list the stack status for an individual switch and for other switches that have been discovered in the same subnet.

Syntax: show stack [candidates | view | all]

Viewing the Status of an Individual Switch. The following example illustrates how to use the CLI in a to display the stack status for that switch. In this case, the switch is in the default stacking configuration.

Syntax: show stack

```
ProCurve(config)# show stack
 Stacking - Stacking Status (This Switch)
  Stack State            : Commander
  Transmission Interval  : 60
  Stack Name             : Big_Waters       Number of members          : 1
  Auto Grab              : Yes              Members unreachable        : 0

  SN MAC Address   System Name       Device Type  Status
  -- ------------- ----------------- -----------  ------------------------
  0  0030c1-7fcc40 3500yl             3500yl      Commander Up
  1  0030c1-7fec40 piles-1            3500yl      Member Up
```

Figure 7-22. Example of Using the Show Stack Command To List the Stacking Configuration for an Individual Switch

Viewing the Status of Candidates the Commander Has Detected.

This example illustrates how to list stack candidates the Commander has discovered in the ip subnet (broadcast domain).

Syntax: show stack candidates

```
ProCurve (config)# show stack candidates
  Stack Candidates
   Candidate MAC System Name                Device Type
   ------------- ------------------------- -----------
   0060b0-889e00 DEFAULT_CONFIG             3500yl
```

Figure 7-23. Example of Using the Show Stack Candidates Command To List Candidates

Viewing the Status of all Stack-Enabled Switches Discovered in the IP Subnet. The next example lists all the stack-configured switches discovered in the IP subnet. Because the switch on which the **show stack all** command was executed is a candidate, it is included in the "Others" category.

Syntax: show stack all

```
ProCurve(config)# show stack all

 Stacking - Stacking Status (All)

  Stack Name       MAC Address   System Name                Status
  ---------------- ------------- -------------------------- -------------
  Big_Waters       0030c1-7fcc40 3500yl                     Commander Up
                   0030c1-7fec40 Big_Waters-1               Member Up
  Others:          0060b0-889e00 DEFAULT_CONFIG             Candidate
```

Figure 7-24. Result of Using the Show Stack All Command To List Discovered Switches in the IP Subnet

Viewing the Status of the Commander and Current Members of the Commander's Stack. The next example lists all switches in the stack of the selected switch.

Syntax: show stack view

```
ProCurve(config)# show stack view
 Stack Members
  SN MAC Address   System Name       Device Type Status
  -- ------------- ----------------- ----------- --------------
  0  0030c1-7fcc40 3500yl             3500yl      Commander Up
  1  0030c1-7fec40 Big_Waters-1       3500yl      Member Up
```

Figure 7-25. Example of the Show Stack View Command To List the Stack Assigned to the Selected Commander

Using the CLI To Configure a Commander Switch

You can configure any stacking-enabled switch to be a Commander as long as the intended stack name does not already exist on the broadcast domain. (When you configure a Commander, you automatically create a corresponding stack.)

Before you begin configuring stacking parameters:

1. Configure IP addressing on the switch intended for stack commander and, if not already configured, on the primary VLAN. (For more on configuring IP addressing, refer to the *Management and Configuration Guide* for your switch.)

Note

The primary VLAN must have an IP address in order for stacking to operate properly. For more on the primary VLAN, see "The Primary VLAN" on page 2-51.

2. Configure a Manager password on the switch intended for commander. (The Commander's Manager password controls access to stack Members.) For more on passwords, see the local manager and operator password information in the *Access Security Guide* for your switch.

Configure the Stack Commander. Assigning a stack name to a switch makes it a Commander and automatically creates a stack.

Syntax: stack commander < *name-str* >

This example creates a Commander switch with a stack name of **Big_Waters**. (Note that if stacking was previously disabled on the switch, this command also enables stacking.)

```
ProCurve(config)# stack commander Big_Waters
```

As the following **show stack** display shows, the Commander switch is now ready to add members to the stack.

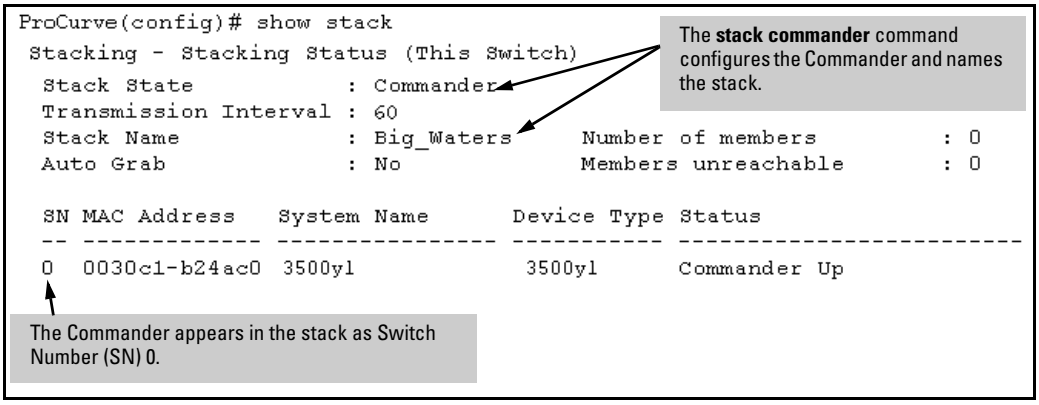

ProCurve(config)# show stack
Stacking - Stacking Status (This Switch)
Stack State : Commander
Transmission Interval : 60
Stack Name : Big_Waters
Number of members : 0
Auto Grab : No
Members unreachable : 0
SN MAC Address System Name Device Type Status
-- ------------ ---------- ---------- ----------
0 0030c1-b24ac0 3500yl 3500yl Commander Up
The stack commander command configures the Commander and names the stack.
The Commander appears in the stack as Switch Number (SN) 0.

Figure 7-26. Example of the Commander's Show Stack Screen with Only the Commander Discovered

Using a Member's CLI to Convert the Member to the Commander of a New Stack. This procedure requires that you first remove the Member from its current stack, then create the new stack. If you do not know the MAC address for the Commander of the current stack, use **show stack** to list it.

Syntax: no stack
stack commander < *stack name* >

Suppose, for example, that a ProCurve switch named "Bering Sea" is a Member of a stack named "Big_Waters". To use the switch's CLI to convert it from a stack Member to the Commander of a new stack named "Lakes", you would use the following commands:

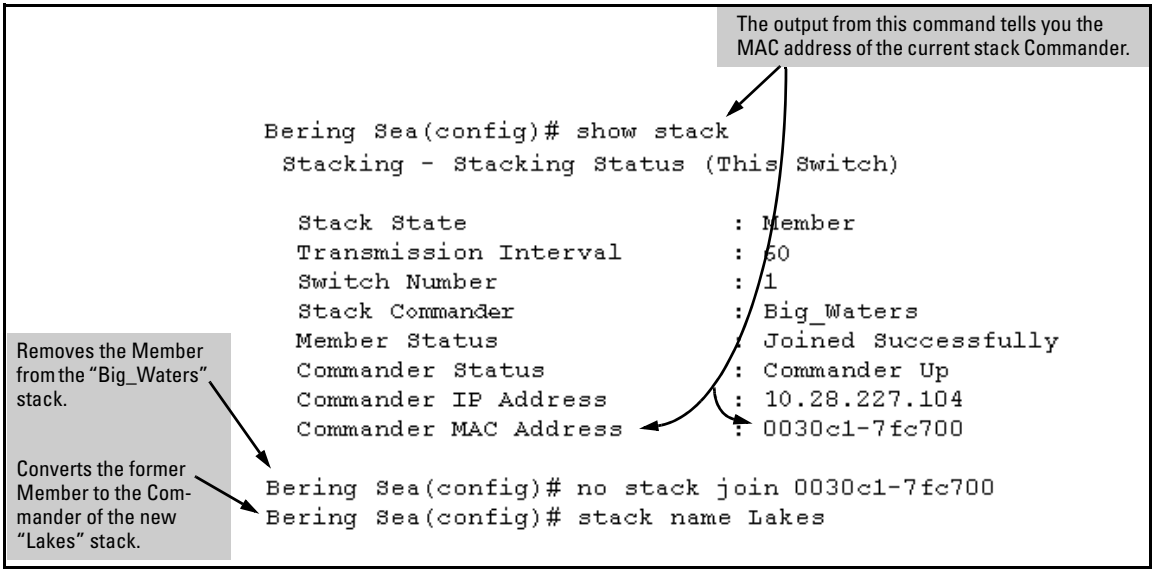

The output from this command tells you the MAC address of the current stack Commander.
Bering Sea(config)# show stack
Stacking - Stacking Status (This Switch)
Stack State : Member
Transmission Interval : 60
Switch Number : 1
Stack Commander : Big_Waters
Member Status : Joined Successfully
Commander Status : Commander Up
Commander IP Address : 10.28.227.104
Commander MAC Address : 0030c1-7fc700
Removes the Member from the "Big_Waters" stack.
Converts the former Member to the Commander of the new "Lakes" stack.
Bering Sea(config)# no stack join 0030c1-7fc700
Bering Sea(config)# stack name Lakes

Figure 7-27. Example of Using a Member's CLI To Convert the Member to the Commander of a New Stack

Adding to a Stack or Moving Switches Between Stacks

You can add switches to a stack by adding discovered Candidates or by moving switches from other stacks that may exist in the same subnet. (You cannot add a Candidate that the Commander has not discovered.)

In its default configuration, the Commander's **Auto-Grab** parameter is set to **No** to give you manual control over which switches join the stack and when they join. This prevents the Commander from automatically trying to add every Candidate it finds that has **Auto Join** set to **Yes** (the default for the Candidate).

(If you want any eligible Candidate to automatically join the stack when the Commander discovers it, configure **Auto Grab** in the Commander to **Yes**. When you do so, *any* Candidate discovered with **Auto Join** set to **Yes** (the default) and no Manager password will join the stack, up to the limit of 15 Members.)

Using the Commander's CLI To Manually Add a Candidate to the Stack. To manually add a candidate, you will use:

- A switch number (**SN**) to assign to the new member. Member SNs range from 1 to 15. To see which SNs are already assigned to Members, use **show stack view**. You can use any SN not included in the listing. (SNs are viewable only on a Commander switch.)
- The MAC address of the discovered Candidate you are adding to the stack. To see this data, use the **show stack candidates** listing .

For example:

```
ProCurve(config)# show stack view
 Stack Members

  SN MAC Address    System Name       Device Type  Status
  -- -------------- ----------------- -----------  --------------------------
  0  0030c1-7fec40 3500yl             3500yl       Commander Up
  1  0060b0-880a80 Indian Ocean       3500yl       Member Up
```

In this stack, the only SNs in use are 0 and 1, so you can use any SN number from 2 through 15 for new Members. (The SN of "0" is always reserved for the stack Commander.)

Note: When manually adding a switch, you must assign an SN. However, if the Commander automatically adds a new Member, it assigns an SN from the available pool of unused SNs.

Figure 7-28. Example of How To Determine Available Switch Numbers (SNs)

To display all discovered Candidates with their MAC addresses, execute **show stack candidates** from the Commander's CLI. For example, to list the discovered candidates for the above Commander:

```
ProCurve(config)# show stack candidates
 Stack Candidates
  Candidate MAC System Name                Device Type
  ------------- -------------------------- -----------
  0030c1-b24ac0 North Sea                   3500yl
  0060b0-df1a00 DEFAULT_CONFIG              3500yl
```

MAC addresses of discovered Candidates.

Figure 7-29. Example of How To Determine MAC Addresses of Discovered Candidates

Knowing the available switch numbers (**SN**s) and Candidate MAC addresses, you can proceed to manually assign a Candidate to be a Member of the stack:

Syntax: stack member < *switch-number* > mac-address < *mac-addr* >
[password < *password-str* >]

For example, if the switch in the above listing did not have a Manager password and you wanted to make it a stack Member with an **SN** of **2**, you would execute the following command:

```
ProCurve(config)# stack member 2 mac-address 0060b0-
dfla00
```

The **show stack view** command then lists the Member added by the above command:

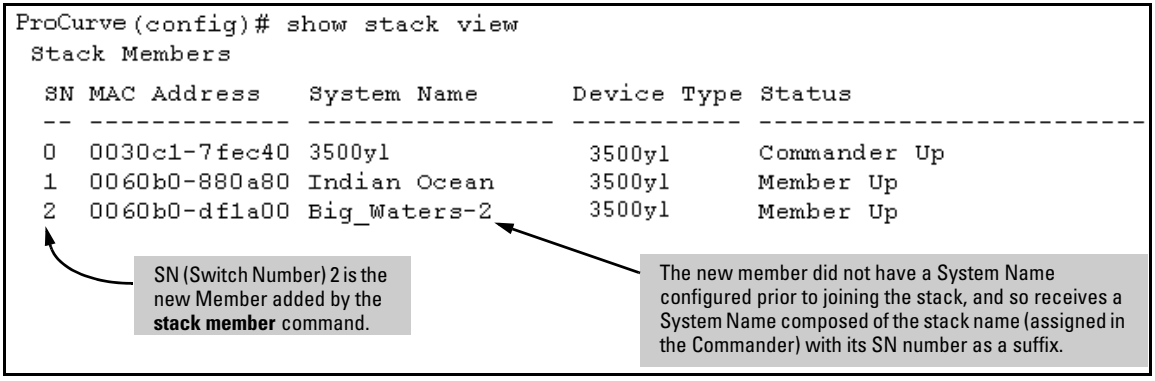

ProCurve(config)# show stack view
Stack Members
SN MAC Address System Name Device Type Status
-- ------------- ---------------- ----------- -------------------------
0 0030c1-7fec40 3500yl 3500yl Commander Up
1 0060b0-880a80 Indian Ocean 3500yl Member Up
2 0060b0-df1a00 Big_Waters-2 3500yl Member Up
SN (Switch Number) 2 is the new Member added by the stack member command.
The new member did not have a System Name configured prior to joining the stack, and so receives a System Name composed of the stack name (assigned in the Commander) with its SN number as a suffix.

Figure 7-30. Example Showing the Stack After Adding a New Member

Using Auto Join on a Candidate. In the default configuration, a Candidate's Auto Join parameter is set to "Yes", meaning that it will automatically join a stack if the stack's Commander detects the Candidate and the Commander's Auto Grab parameter is set to "Yes". You can disable Auto Join on a Candidate if you want to prevent automatic joining in this case. There is also the instance where a Candidate's Auto Join is disabled, for example, when a Commander leaves a stack and its members automatically return to Candidate status, or if you manually remove a Member from a stack. In this case, you may want to reset Auto Join to "Yes".

Status: **[no] stack auto-join**

```
ProCurve(config)# no stack auto-join
```
Disables Auto Join on a Candidate.

```
ProCurve(config)# stack auto-join
```
Enables Auto Join on a Candidate.

Using a Candidate CLI To Manually "Push" the Candidate Into a Stack . Use this method if any of the following apply:

- The Candidate's **Auto Join** is set to **Yes** (and you do not want to enable **Auto Grab** on the Commander) or the Candidate's **Auto Join** is set to **No**.
- Either you know the MAC address of the Commander for the stack into which you want to insert the Candidate, or the Candidate has a valid IP address and is operating in your network.

Syntax: stack join < *mac-addr* >

where: ***< mac-addr >*** is the MAC address of the Commander in the destination stack.

Use Telnet (if the Candidate has an IP address valid for your network) or a direct serial port connection to access the CLI for the Candidate switch. For example, suppose that a Candidate named "North Sea" with **Auto Join** off and a valid IP address of 10.28.227.104 is running on a network. You could Telnet to the Candidate, use **show stack all** to determine the Commander's MAC address, and then "push" the Candidate into the desired stack.

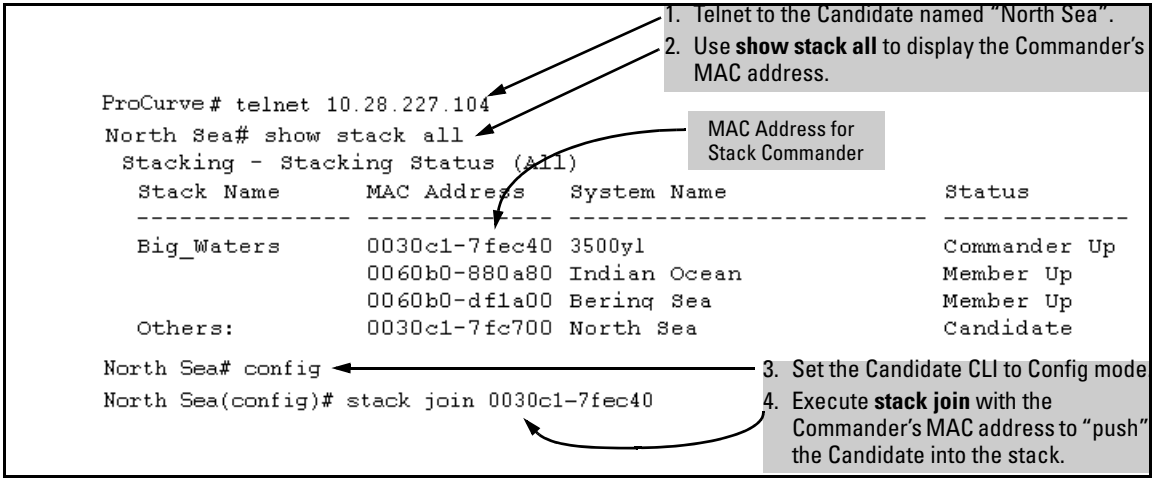

1. Telnet to the Candidate named "North Sea".
2. Use show stack all to display the Commander's MAC address.
ProCurve# telnet 10.28.227.104
North Sea# show stack all
MAC Address for Stack Commander
Stacking - Stacking Status (All)
Stack Name MAC Address System Name Status
Big_Waters 0030c1-7fec40 3500yl Commander Up
0060b0-880a80 Indian Ocean Member Up
0060b0-df1a00 Bering Sea Member Up
Others: 0030c1-7fc700 North Sea Candidate
North Sea# config
3. Set the Candidate CLI to Config mode
North Sea(config)# stack join 0030c1-7fec40
4. Execute stack join with the Commander's MAC address to "push" the Candidate into the stack.

Figure 7-31. Example of "Pushing" a Candidate Into a Stack

To verify that the Candidate successfully joined the stack, execute **show stack all** again to view the stacking status.

Using the Destination Commander CLI To "Pull" a Member from Another Stack. This method uses the Commander in the destination stack to "pull" the Member from the source stack.

Syntax: stack member < *switch-number* >
mac-address < *mac-addr* >
[password < *password-str* >]

In the destination Commander, use **show stack all** to find the MAC address of the Member you want to pull into the destination stack. For example, suppose you created a new Commander with a stack name of "Cold_Waters" and you wanted to move a switch named "Bering Sea" into the new stack:

```
ProCurve(config)# show stack all
 Stacking - Stacking Status (All)
  Stack Name      MAC Address   System Name                Status
  --------------- ------------- -------------------------- -------------
  Big_Waters      0030c1-7fec40 3500yl                     Commander Up
                  0060b0-880a80 Indian Ocean               Member Up
                  0060b0-df1a00 Bering Sea                 Member Up
  Cold_Waters     0030c1-7fc700 3500yl                     Commander Up
```

Move this switch into the "Cold Waters" stack.

Figure 7-32. Example of Stack Listing with Two Stacks in the Subnet

You would then execute the following command to pull the desired switch into the new stack:

```
ProCurve(config)# stack member 1 mac-address 0060b0-
df1a00
```

Where **1** is an unused switch number (**SN**).

Since a password is not set on the Candidate, a password is not needed in this example.

You could then use **show stack all** again to verify that the move took place.

Using a Member CLI To "Push" the Member into Another Stack. You can use the Member's CLI to "push" a stack Member into a destination stack if you know the MAC address of the destination Commander.

Syntax: stack join <*mac-addr*>

where: ***< mac-addr >*** is the MAC address of the Commander for the destination stack.

Converting a Commander to a Member of Another Stack. Removing the Commander from a stack eliminates the stack and returns its Members to the Candidate pool with **Auto Join** disabled.

Syntax: no stack name < *stack name*>
stack join < *mac-address* >

If you don't know the MAC address of the destination Commander, you can use **show stack all** to identify it.

For example, suppose you have a switch operating as the Commander for a temporary stack named "Test". When it is time to eliminate the temporary "Test" stack and convert the switch into a member of an existing stack named "Big_Waters", you would execute the following commands in the switch's CLI:

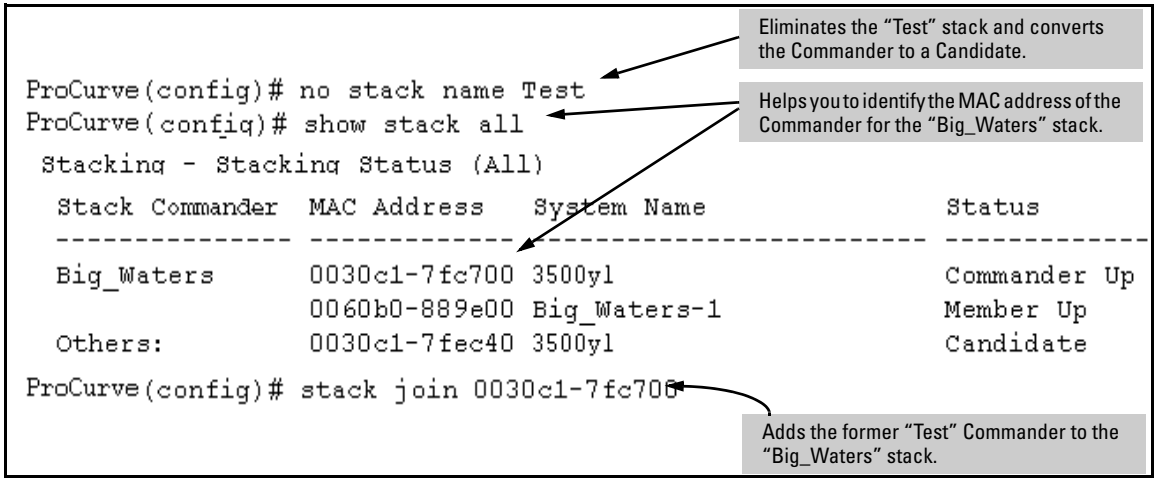

Eliminates the "Test" stack and converts the Commander to a Candidate.
Helps you to identify the MAC address of the Commander for the "Big_Waters" stack.
Adds the former "Test" Commander to the "Big_Waters" stack.
ProCurve(config)# no stack name Test
ProCurve(config)# show stack all
Stacking - Stacking Status (All)
Stack Commander MAC Address System Name Status
Big_Waters 0030c1-7fc700 3500yl Commander Up
0060b0-889e00 Big_Waters-1 Member Up
Others: 0030c1-7fec40 3500yl Candidate
ProCurve(config)# stack join 0030c1-7fc700

Figure 7-33. Example of Command Sequence for Converting a Commander to a Member

Using the CLI To Remove a Member from a Stack

You can remove a Member from a stack using the CLI of either the Commander or the Member.

Note

When you remove a Member from a stack, the Member's **Auto Join** parameter is set to **No**.

Using the Commander CLI To Remove a Stack Member. This option requires the switch number (SN) and the MAC address of the switch to remove. (Because the Commander propagates its Manager password to all stack members, knowing the Manager password is necessary only for gaining access to the Commander.)

Syntax: [no] stack member <*switch-num*> mac-address <*mac-addr*>

Use **show stack view** to list the stack Members. For example, suppose that you wanted to use the Commander to remove the "North Sea" Member from the following stack:

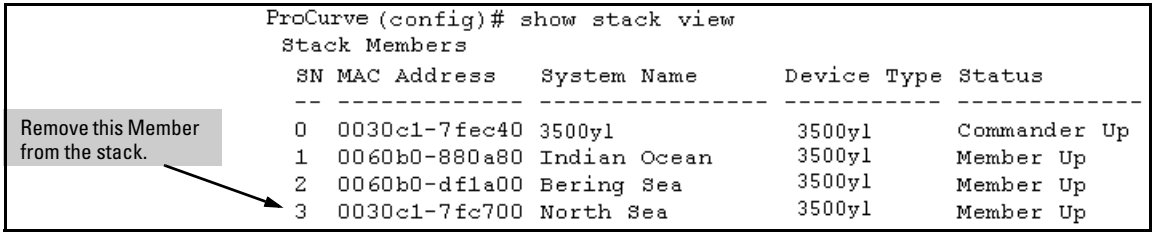

Remove this Member from the stack.
ProCurve(config)# show stack view
Stack Members
SN MAC Address System Name Device Type Status
-- ------------ ---------------- ----------- -------------
0 0030c1-7fec40 3500yl 3500yl Commander Up
1 0060b0-880a80 Indian Ocean 3500yl Member Up
2 0060b0-df1a00 Bering Sea 3500yl Member Up
3 0030c1-7fc700 North Sea 3500yl Member Up

Figure 7-34. Example of a Commander and Three Switches in a Stack

You would then execute this command to remove the "North Sea" switch from the stack:

```
ProCurve(config)# no stack member 3 mac-address 0030c1-
7fc700
```

where:

- **3** is the "North Sea" Member's switch number (**SN**)
- **0030c1-7fc700** is the "North Sea" Member's MAC address

Using the Member's CLI To Remove the Member from a Stack.

Syntax: no stack join *<mac-addr>*

To use this method, you need the Commander's MAC address, which is available using the show stack command in the Member's CLI. For example:

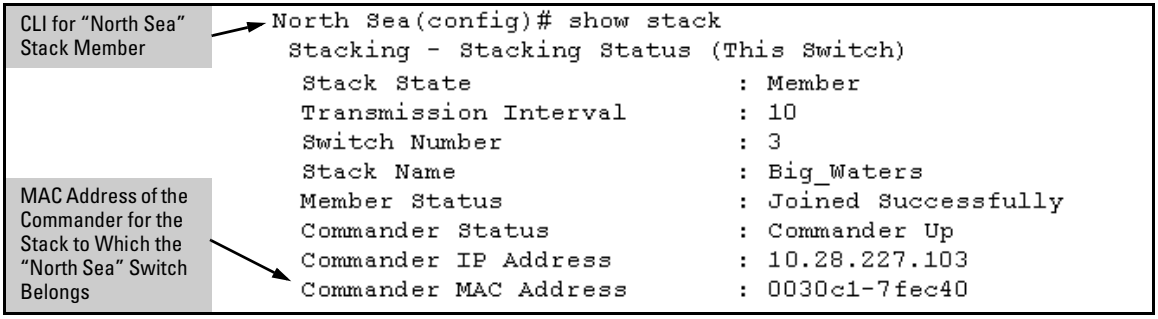

CLI for "North Sea" Stack Member
MAC Address of the Commander for the Stack to Which the "North Sea" Switch Belongs
North Sea(config)# show stack
Stacking - Stacking Status (This Switch)
Stack State : Member
Transmission Interval : 10
Switch Number : 3
Stack Name : Big_Waters
Member Status : Joined Successfully
Commander Status : Commander Up
Commander IP Address : 10.28.227.103
Commander MAC Address : 0030c1-7fec40

Figure 7-35. Example of How To Identify the Commander's MAC Address from a Member Switch

You would then execute this command in the "North Sea" switch's CLI to remove the switch from the stack:

```
North Sea(config)# no stack join 0030c1-7fec40
```

Using the CLI To Access Member Switches for Configuration Changes and Traffic Monitoring

After a Candidate becomes a Member, you can use the telnet command from the Commander to access the Member's CLI or console interface for the same configuration and monitoring that you would do through a Telnet or direct-connect access from a terminal.

Syntax: telnet *<switch-number>*

where: unsigned integer is the switch number (**SN**) assigned by the Commander to each member (range: **1** - **15**).

To find the switch number for the Member you want to access, execute the **show stack view** command in the Commander's CLI. For example, suppose that you wanted to configure a port trunk on the switch named "North Sea" in the stack named "Big_Waters". Do do so you would go to the CLI for the "Big_Waters" Commander and execute show stack view to find the switch number for the "North Sea" switch:

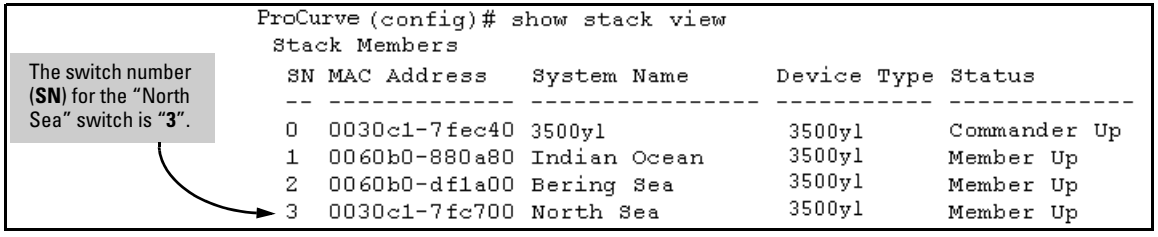

The switch number (SN) for the "North Sea" switch is "3".
ProCurve(config)# show stack view
Stack Members
SN MAC Address System Name Device Type Status
-- ------------ ---------------- ----------- -------------
0 0030c1-7fec40 3500yl 3500yl Commander Up
1 0060b0-880a80 Indian Ocean 3500yl Member Up
2 0060b0-df1a00 Bering Sea 3500yl Member Up
3 0030c1-7fc700 North Sea 3500yl Member Up

Figure 7-36. Example of a Stack Showing Switch Number (SN) Assignments

To access the "North Sea" console, you would then execute the following **telnet** command:

```
ProCurve(config)# telnet 3
```

You would then see the CLI prompt for the "North Sea" switch, allowing you to configure or monitor the switch as if you were directly connected to the console.

SNMP Community Operation in a Stack

Community Membership

In the default stacking configuration, when a Candidate joins a stack, it automatically becomes a Member of any SNMP community to which the Commander belongs, even though any community names configured in the Commander are not propagated to the Member's SNMP Communities listing. However, if a Member has its own (optional) IP addressing, it can belong to SNMP communities to which other switches in the stack, including the Commander, do not belong. For example:

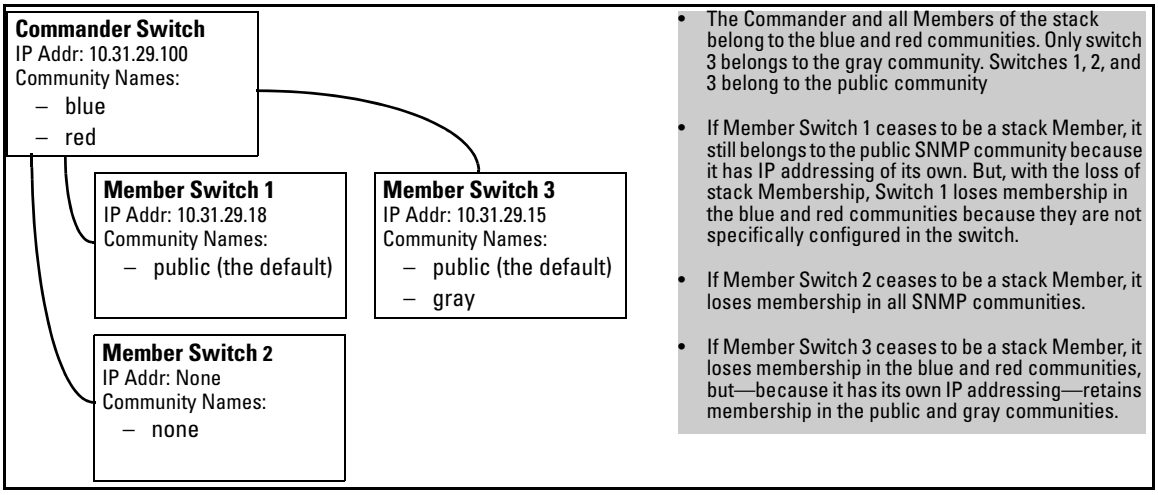

Commander Switch
IP Addr: 10.31.29.100
Community Names:
– blue
– red
Member Switch 1
IP Addr: 10.31.29.18
Community Names:
– public (the default)
Member Switch 3
IP Addr: 10.31.29.15
Community Names:
– public (the default)
– gray
Member Switch 2
IP Addr: None
Community Names:
– none
• The Commander and all Members of the stack belong to the blue and red communities. Only switch 3 belongs to the gray community. Switches 1, 2, and 3 belong to the public community
• If Member Switch 1 ceases to be a stack Member, it still belongs to the public SNMP community because it has IP addressing of its own. But, with the loss of stack Membership, Switch 1 loses membership in the blue and red communities because they are not specifically configured in the switch.
• If Member Switch 2 ceases to be a stack Member, it loses membership in all SNMP communities.
• If Member Switch 3 ceases to be a stack Member, it loses membership in the blue and red communities, but—because it has its own IP addressing—retains membership in the public and gray communities.

Figure 7-37. Example of SNMP Community Operation with Stacking

SNMP Management Station Access to Members Via the Commander.

To use a management station for SNMP Get or Set access through the Commander's IP address to a Member, you must append **@sw<*switch number*>** to the community name. For example, in figure 7-37, you would use the following command in your management station to access Switch 1's MIB using the blue community:

```
snmpget <MIB variable> 10.31.29.100 blue@sw1
```

Note that because the gray community is only on switch 3, you could not use the Commander IP address for gray community access from the management station. Instead, you would access switch 3 directly using the switch's own IP address. For example:

```
snmpget <MIB variable> 10.31.29.15 gray
```

Note that in the above example (figure 7-37) you cannot use the public community through the Commander to access any of the Member switches. For example, you can use the public community to access the MIB in switches 1 and 3 by using their unique IP addresses. However, you must use the red or blue community to access the MIB for switch 2.

```
snmpget <MIB variable> 10.31.29.100 blue@sw2
```

Using the CLI To Disable or Re-Enable Stacking

In the default configuration, stacking is enabled on the switch. You can use the CLI to disable stacking on the switch at any time. Disabling stacking has the following effects:

- **Disabling a Commander:** Eliminates the stack, returns the stack Members to Candidates with **Auto Join** disabled, and changes the Commander to a stand-alone (nonstacking) switch. You must re-enable stacking on the switch before it can become a Candidate, Member, or Commander.
- **Disabling a Member:** Removes the Member from the stack and changes it to a stand-alone (nonstacking) switch. You must re-enable stacking on the switch before it can become a Candidate, Member, or Commander.
- **Disabling a Candidate:** Changes the Candidate to a stand-alone (nonstacking) switch.

Syntax: no stack (*Disables stacking on the switch.*)
stack (*Enables stacking on the switch.*)

Transmission Interval

All switches in the stack must be set to the same transmission interval to help ensure proper stacking operation. HP recommends that you leave this parameter set to the default 60 seconds.

Syntax: stack transmission-interval < *seconds* >

Stacking Operation with Multiple VLANs Configured

Stacking uses the primary VLAN in a switch. In the factory-default configuration, the DEFAULT_VLAN is the primary VLAN. However, you can designate any VLAN configured in the switch as the primary VLAN. (See "The Primary VLAN" on page 2-51.)

When using stacking in a multiple-VLAN environment, the following criteria applies:

- Stacking uses only the primary VLAN on each switch in a stack.
- The primary VLAN can be tagged or untagged as needed in the stacking path from switch to switch.
- The same VLAN ID (VID) must be assigned to the primary VLAN in each stacked switch.

Status Messages

Stacking screens and listings display these status messages:

Message	Condition	Action or Remedy
Candidate Auto-join	Indicates a switch configured with Stack State set to **Candidate, Auto Join** set to **Yes** (the default), and no Manager password.	None required
Candidate	Candidate cannot automatically join the stack because one or both of the following conditions apply: • Candidate has **Auto Join** set to **No**. • Candidate has a Manager password.	Manually add the candidate to the stack.
Commander Down	Member has lost connectivity to its Commander.	Check connectivity between the Commander and the Member.
Commander Up	The Member has stacking connectivity with the Commander.	None required.
Mismatch	This may be a temporary condition while a Candidate is trying to join a stack. If the Candidate does not join, then stack configuration is inconsistent.	Initially, wait for an update. If condition persists, reconfigure the Commander or the Member.
Member Down	A Member has become detached from the stack. A possible cause is an interruption to the link between the Member and the Commander.	Check the connectivity between the Commander and the Member.
Member Up	The Commander has stacking connectivity to the Member.	None required.
Rejected	The Candidate has failed to be added to the stack.	The candidate may have a password. In this case, manually add the candidate. Otherwise, the stack may already be full. A stack can hold up to 15 Members (plus the Commander).

8

QinQ (Provider Bridging)

Contents

Overview 8-3

Introduction 8-4

How QinQ Works 8-5

Features and Benefits 8-5

Terminology 8-6

Operating Rules and Guidelines 8-7

Enabling QinQ and Configuring QinQ Modes 8-7

QinQ Mixed Vlan Mode 8-8

Configuring VLANs 8-8

Operating Notes and Restrictions 8-10

Configuring QinQ 8-13

General Configuration Steps 8-13

Enabling QinQ 8-14

Setting up S-VLANs 8-14

Configuring Per-Port S-VLAN Membership 8-15

Configuring Port-Types 8-16

Configuration Example 8-17

Updating QinQ Configurations 8-22

Changing QinQ Modes 8-22

Disabling QinQ 8-22

Changing VLAN Port Memberships (Mixed Vlan Mode) 8-22

Moving Ports between C-VLANs and S-VLANs (Mixed Vlan Mode) . 8-23

Displaying QinQ Config and Status 8-24

Show Commands for QinQ 8-24

Show Commands for VLANs 8-25
Displaying Spanning Tree Status 8-27
Effects of QinQ on Other Switch Features 8-28
Event Log Messages and SNMP Support 8-33
SNMP Support and MIB Objects 8-33

Overview

This chapter describes how to enable QinQ operations on the switch and how to configure provider bridge S-VLANs and port assignments.

For information on how to configure and use static, port-based and protocol-based VLANs, refer to chapter 2, “Static Virtual LANs (VLANs)”.

For information on how to configure dynamic VLANs on the switch, refer to chapter 3, “GVRP”.

License Requirements

In the 3500yl, 5400zl, 6600 and 8200zl switches, QinQ is included with the Premium License. In the 6200yl switches, this feature is included with the base feature set.

Introduction

The IEEE 802.1ad specification, commonly known as QinQ or provider bridging, extends the IEEE 802.1Q standard by providing for a second tier of VLANs in a bridged network. The general purpose of QinQ is to allow frames from multiple customers to be forwarded (or tunneled) through another topology (provider network) using service VLANs or S-VLANs. The provider bridge, which may comprise multiple devices in the service provider domain, looks like a simple bridge port to the customers traffic and maintains the customer's VLANs.

Figure 8-1 shows a sample QinQ topology and use model. Customer A has LANs spread across multiple site locations and may want to link them together in a single logical LAN. To do this, the customer could have a cable laid out for the entire distance interconnecting the three sites. A more cost-effective and scalable alternative, however, would be to tunnel frames through the provider's network to interconnect all the sites subscribing to the service. This solution can be delivered using QinQ.

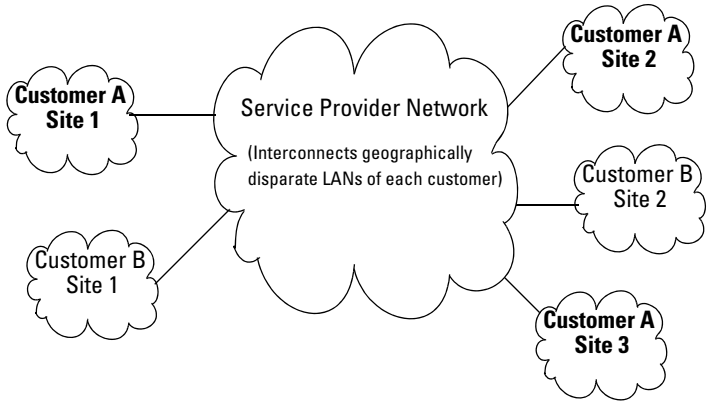


Figure 8-1. QinQ Network Diagram

Note

The so-called 'Service Provider' and 'Customers' may belong to the same business entity, as in the case where a single enterprise uses QinQ to help segregate local networks and increase the scalability of their backbone infrastructure.

How QinQ Works

Under QinQ, the provider network operates on a different VLAN space, independent of the VLANs that are used in the customer network as shown in Figure 8-2.

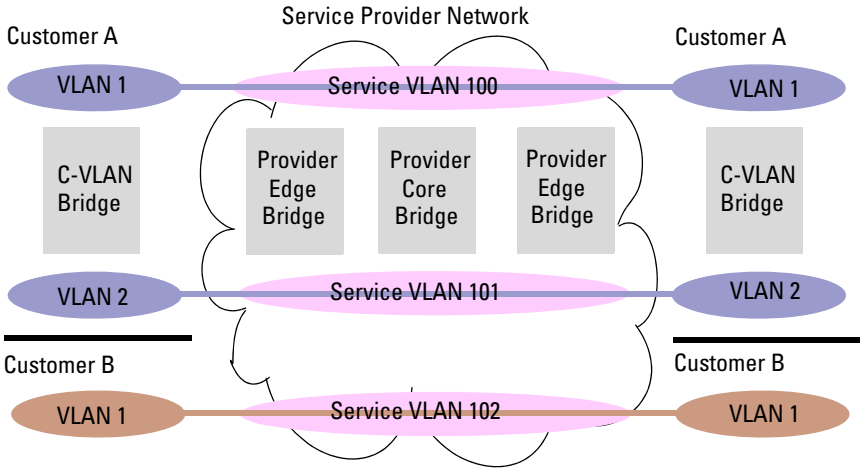


Figure 8-2. Example of VLANs in a QinQ Configuration

Customer VLANs (referred to as C-VLANs by the IEEE 802.1ad specification) are not used to make any forwarding decisions inside the provider network where customer frames get assigned to service VLANs (S-VLANs). Inside the provider cloud, frames are forwarded based on the S-VLAN tag only, while the C-VLAN tag remains shielded during data transmission. The S-VLAN tag is removed when the frame exits the provider network, restoring the original customer frame.

Features and Benefits

- Increases the VLAN space in a provider network or enterprise backbone.
- Reduces the number of VLANs that a provider needs to support within the provider network for the same number of customers.
- Enables customers to plan their own VLAN IDs, without running into conflicts with service provider VLAN IDs.
- Provides a simple Layer 2 VPN solution for small-sized MANs (Metropolitan Area Networks) or intranets.
- Provides for customer traffic isolation at Layer 2 within a Service Provider network.

Terminology

C-VLANs. Customer network VLANs that can exist across multiple locations. These are assigned and managed by each customer and are local to the customer space.

C-VLAN bridge. A customer-owned device operating regular 802.1Q VLANs.

Customer. The consumer of network services delivered by a service provider.

Customer-network port. Customer-facing port on a provider edge device. The equivalent of 'CN' ports of the IEEE 802.1ad standard.

Customer VLAN. See **C-VLAN**.

IEEE 802.1ad. Specification that allows a service provider to assign a unique VLAN identifier (called the Service VLAN ID or S-VID) to customers using multiple VLANs, thereby extending the total number of VLANs that can be supported within the provider network.

Mixed vlan mode device. Device that supports both C-VLANs and S-VLANs. A device configured in **qinq mixedvlan** mode can do regular CVLAN switching/routing (standard bridge behavior) and can also serve as a provider edge device tunneling frames into and out of the provider network.

Port-based interface. Untagged customer-network ports or trunks on a QinQ enabled device. See also **S-tagged interface**.

Provider-network port. Port on an S-VLAN bridge that connects to the provider network. This equates to 'PN' ports of the IEEE 802.1ad standard.

QinQ. A feature that enables service providers to use a single VLAN-ID to support multiple customer VLANs by encapsulating the 802.1Q VLAN tag within another 802.1Q frame. See also **IEEE 802.1ad**.

Service Provider. The provider of the network that provides one or more service instances to a customer.

S-tagged interface. Tagged customer-network ports or trunks on a QinQ enabled device. See also **port-based interface**.

Service VLAN. See **S-VLAN**.

S-VLAN. Service VLANs that are used to tunnel customer frames through the provider network to customer sites. These are managed by the service provider who can assign each customer a unique S-VLAN ID.

S-VLAN bridge. Provider-owned device configured in **qinq svlan** mode that uses S-VLANs only to forward frames in the provider network. This bridge can be further sub-classified as:

- **Provider edge bridge**. An svlan bridge that has customer network ports configured on the device.
- **Provider core bridge**. A device in the core of the provider network that does not interface with any customer bridges. All ports on the device are provider network ports, and the S-VLAN bridge only receives and forwards S-tagged frames.

Tunnel VLAN. See **S-VLAN**.

Operating Rules and Guidelines

This section provides an overview of QinQ operations and restrictions on the switch. For details of CLI commands and configuration procedures, refer to "Configuring QinQ" on page 8-13.

Enabling QinQ and Configuring QinQ Modes

By default, QinQ is disabled. When QinQ is enabled via the CLI, an operating mode is globally configured on the switch. Two QinQ modes are supported:

- **qinq mixedvlan**: C-VLANs and S-VLANs are both supported, with regular switching/routing based on C-VLAN tags in the C-VLAN domain, while S-VLANs are used for QinQ tunneling through the provider network.
- **qinq svlan**: C-VLANs are NOT supported on the device. All configured VLANs on the switch must be S-VLANs.

Table 8-1 shows how the various QinQ modes and operations impact VLAN configuration options on the switch.

Table 8-1. Relationship of QinQ Operating Modes to VLAN Environments

QinQ Operation	CLI Command	VLAN Options
QinQ disabled		
No QinQ support (Default)	**no qinq**	Only regular VLAN commands are available. If QinQ is disabled, S-VLAN commands are not available.
QinQ enabled		
QinQ mixed vlan mode	**qinq mixedvlan**	Both S-VLAN and regular VLAN commands (known as C-VLANs in a mixed vlan environment) are available.
QinQ svlan mode	**qinq svlan**	No regular VLAN commands are available. All VLANs configured on the switch are S-VLANs only.

QinQ Mixed Vlan Mode

The QinQ mixed vlan mode configuration supports both C-VLAN and S-VLAN operations on the same device. This allows the use of S-VLAN member ports for QinQ tunneling, while regular ports can still do switching or routing within the C-VLAN space. To tunnel customer frames through the provider network, you can externally connect a regular port to a customer-network port, eliminating the need for a separate S-VLAN bridge device to perform such operations. When configuring VLANs on a mixed vlan mode device, a separate **svlan <vid>** command is used to distinguish the S-VLAN type from regular VLANs.

The main advantage for QinQ mixed vlan mode is that users do not have to dedicate the entire switch as a QinQ access switch. For a high density chassis switch such as the 5400zl or 8200zl series, customers can use regular ports for normal LAN switching, while S-VLAN member ports can be configured to access the QinQ provider network (see Figure 8-3). There are some additional restrictions in mixed-VLAN mode (see "Operating Notes and Restrictions" on page 8-10 for details).

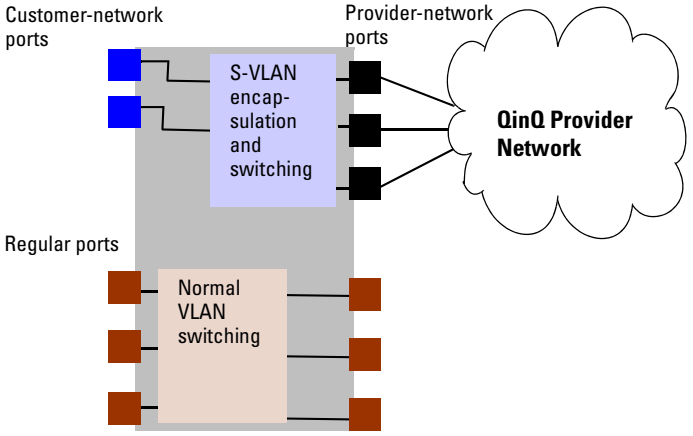


Figure 8-3. ProCurve Switch in Mixed-VLAN mode

Configuring VLANs

- A VLAN created on a QinQ mixed vlan mode device can be either a regular VLAN (C-VLAN) or a tunnel VLAN (S-VLAN). C-VLANs have no mapping/relation whatsoever to the S-VLANs on the device.
- VLANs created on a QinQ svlan mode device can be S-VLANs only. S-VLANs provide QinQ tunneling of customer frames and behave like a port-based/s-tagged interface (see "Setting up S-VLANs" on page 8-14 for configuration details).

QinQ and Duplicate VIDs. Duplicate VID's for c-tagged and s-tagged VLANs (for example, C-VID=100; S-VID=100) are allowed in certain cases and disallowed in others. Customer-network ports are essentially S-VLAN ports: they simply read the C-tags in the customer frame to insert them into the appropriate untagged S-VLAN for that port. Once this double-tagging occurs, frames are forwarded based on the S-VLAN tag only, while the C-VLAN tag remains shielded during data transmission. See Figure 8-4 for examples of allowed configurations.

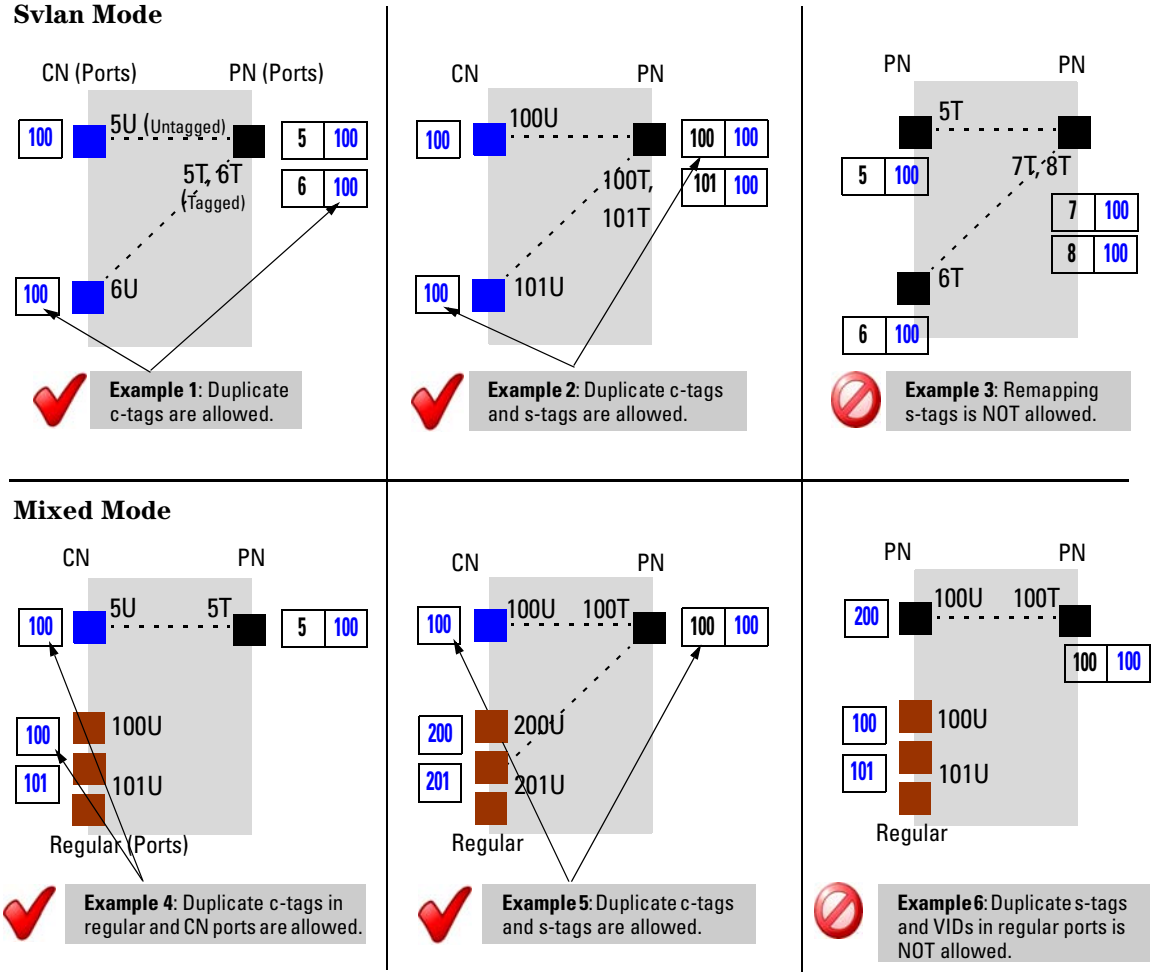


Figure 8-4. QinQ and Duplicate VIDs: Examples of Allowed Configurations

Assigning Ports to VLANs. In mixed vlan mode, a port can be a member of a C-VLAN or of an S-VLAN but not both. For details, on assigning membership to provider-based VLANs, see "Configuring Per-Port S-VLAN Membership" on page 8-15.

Configuring Port-Types. The IEEE 802.1ad standard requires that every svlan member port be configured as either a provider-network or as a customer-network port. In a typical deployment scenario, customer-network ports will be configured as untagged members of S-VLANs while provider-network ports will be configured as tagged members of S-VLANs. Note the following configuration rules and guidelines:

- All ports of a device that is QinQ enabled (in svlan mode or mixed vlan mode) are provider-network ports by default—if there are any ports that connect to a customer device, they must be manually configured as customer-network ports.
- Configuring a port-type is only applicable if the device is QinQ enabled and the port is a member of an svlan. In QinQ mixed mode, ports that are members of C-VLANs cannot be configured to any port-type.

For more information, see "Configuring Port-Types" on page 8-16.

Note

If a device running in QinQ svlan mode has one or more 'customer-network' ports, it is considered to be a provider edge and not a provider core bridge. This may impact certain operations, such as meshing, UDLD, and stacking. This is because at the edge of the provider network such proprietary protocol are filtered out at customer network ports. This prevents the intermix of stacking/meshing/udld protocols in the customer and provider domains (since they use the same dst-mac address in either domain).

Operating Notes and Restrictions

- **Changing bridge modes requires a reboot**. When changing the operating mode (to/from: QinQ svlan mode, QinQ mixed vlan mode, or QinQ disabled), you will prompted to restart the system before the changes can take effect. Upon reboot, *all configuration information for the prior QinQ mode will be lost.* Any configurations created will be erased, and the device will boot up with a default configuration for the new QinQ mode.
- **Provider edge devices at Layer 2 only**. QinQ does not provide Layer 3 capabilities of complete network isolation between customers. In a mixed VLAN configuration, there is no switching/routing between C-VLANs and S-VLANs. S-VLANs are essentially Layer 2 VLANs that switch packets based on S-VIDs.

- **IP support**. Regular VLANs support IP and can be routing enabled. S-VLANs of mixed vlan mode devices cannot be ip enabled. S-VLANs of svlan mode devices can be ip enabled, though routing related features (such as ip routing) are not supported.
- **Double-tagging causes frame size increases**. Since there is both a provider VLAN tag and customer VLAN tag in each QinQ frame, the size of each 'double-tagged' frame increases by 4 bytes. To accommodate the frame size increase, ProCurve recommends that you configure all port-based S-VLANs to accept jumbo frames. See the section on "Jumbo Frames" in the *Management and Configuration Guide* for details.
- **S-VLAN configuration restrictions:**
  - S-VLAN commands are not available when QinQ is disabled on the switch.
- **VLAN configuration restrictions in mixed vlan mode**:
  - Both C-VLANs and S-VLANs can be configured on the switch. In a mixed mode device, the default VLAN is always a C-VLAN.
  - VLAN types cannot be updated dynamically. A VLAN can only be classified as an S-VLAN or a C-VLAN at the time its created. Once created, the VLAN cannot be moved between being a C-VLAN and an S-VLAN. If a VID that was initially created as a regular VLAN needs to be used for an S-VLAN, the VID must be deleted and recreated as an S-VLAN.
  - If a VLAN being configured as an S-VLAN already exists as a GVRP C-VLAN or a static C-VLAN on the switch, the S-VLAN creation is blocked. Similarly, a C-VLAN creation is blocked if the same VID exists as a static S-VLAN on the device.
  - S-VLANs in a mixed vlan device cannot be configured as a voice-VLAN, primary-VLAN, or management-VLAN.
  - S-VLANs cannot be configured with ip-layer functionality, except for ip-acls.
- **VLAN configuration restrictions in svlan mode**:
  - Only S-VLANs are supported—the keyword on all vlan-related command syntax changes from **vlan** to **svlan**.
  - Routing related features such as ip-routing, RIP, OSPF, PIM, and VRRP are NOT supported in svlan mode.
- **Port-based restrictions**:
  - In QinQ mixed vlan mode, a port must be explicitly GVRP-disabled before it can be assigned to the S-VLAN space (see page 8-15 for details).

- In QinQ mixed vlan mode, only ports that are members of S-VLANs can be configured as customer network or provider network ports; ports that are members of C-VLANs cannot be configured to any port-type.
- QinQ mixed vlan mode devices cannot be connected in an S-VLAN mesh topology. This is because STP cannot be run in the S-VLAN space, and so a mesh topology (or the presence of any redundant links) would result in loops.
- A port can only either be a member of S-VLANs or C-VLANs, but not a combination of both.
- A port cannot be configured as a 'Customer-Edge' as specified in Section 12.13.3 of the IEEE 802.1ad specification. In the current software release, such 'C-tagged interfaces' are not supported—only port-based/S-tagged interfaces are supported.
- Moving ports between C-VLANs and S-VLANs may cause conflicts. For example, if a port has any mirroring/monitoring sessions set up, they will not be allowed to change VLAN domains until these sessions are unconfigured. Refer to "Changing VLAN Port Memberships (Mixed Vlan Mode)" on page 8-22 for additional details.

■ **Interoperating with Other Vendor Devices**. When enabling QinQ, you can configure a unique tpid value, such as 0x8100, to allow the device to interoperate with devices that require this value for the inner and outer VLAN-tag. If the provider tag-type is configured as 0x8100, then:

- customer-network ports cannot be configured as tagged-SVLAN members; and
- tagged-SVLAN members cannot be configured as customer-network ports.

■ **Configuring QinQ with Other Network Protocols**. The networks for both the customer and provider can be complex. For information on how QinQ may impact other network protocols (such as spanning tree, LLDP, and GVRP), refer to "Effects of QinQ on Other Switch Features" on page 8-28.

Configuring QinQ

QinQ must be configured on all the devices and ports participating in the provider bridge. Typically, customer facing ports are configured as untagged members of S-VLANs and provider facing ports are configured as tagged members of S-VLANs. Per the IEEE 802.1ad specification, there is no condition binding port types (customer or provider) to untagged or tagged S-VLAN memberships. Therefore, when configuring QinQ tunnelling on the switch, you would first configure per-port S-VLAN membership (tagged or untagged), and then configure the port type as 'customer-network' or 'provider-network' depending on the device to which the switch port is connected.

Note

A customer-network port can receive S-VLAN tagged frames if the customer and provider agree upon the S-VID association for that customer and the customer device is capable of sending S-VLAN tagged frames.

General Configuration Steps

To configure QinQ, you would take the following steps on all participating provider switches (see the following sections for details and refer to page 8-17 for a configuration example):

1. Enable QinQ on the device, selecting the appropriate qinq mode (svlan or mixed vlan mode).
2. Save the configuration and reboot the switch.
3. Configure S-VLANs and assign per port VLAN membership.
4. Configure port-types for all of the switch ports that carry QinQ traffic across the network.
5. (Optional) Verify the configuration (see "Displaying QinQ Config and Status" on page 8-24).

Caution

A reboot is required to enable/disable QinQ operations on the switch. When moving between qinq modes (**qinq mixedvlan** to **qinq svlan** or vice versa), the switch boots up with a default configuration for the new qinq mode and the configuration parameters of the current mode will be erased out. Refer to "Updating QinQ Configurations" on page 8-22 for details.

Enabling QinQ

By default, QinQ is disabled on the switch. To enable QinQ, the switch must be put into either QinQ mixed vlan mode or QinQ svlan mode by issuing one of the following commands from configuration mode on the CLI.

Syntax: qinq mixedvlan <tag-type[tpid]>

From config mode, globally enables QinQ mixed mode, an environment that supports both S-VLAN and C-VLAN traffic on the same device. This command requires a reboot to take effect.
Default: Disabled.

Syntax: qinq svlan <tag-type[tpid]>

From config mode, globally enables QinQ svlan mode, an S-VLAN only environment that supports supports port-based or s-tagged interfaces of the standard. Requires a reboot to take effect.
Default: Disabled.

Setting up S-VLANs

S-VLANs can be created via the CLI using the **svlan** <vid> command.

Syntax: svlan < *vid* / *ascii-name-string* >
[no] svlan < *vid* >

If < vid > does not exist in the switch, this command creates a port-based S-VLAN with the specified < vid >. If the command does not include options, the CLI moves to the newly created S-VLAN context. If you do not specify an optional name, the switch assigns a name in the default format: **svlan*n*** *where* ***n*** *is the < vid > assigned to the S-VLAN. If the S-VLAN already exists and you enter either the* ***vid*** *or the* ***ascii-name-string****, the CLI moves to the specified S-VLAN's context.*

The **[no]** *form of the command deletes the S-VLAN as follows:*

- *If one or more ports belong only to the S-VLAN to be deleted, the CLI notifies you that these ports will be moved to the default VLAN and prompts you to continue the deletion. For member ports that also belong to another S-VLAN, there is no "move" prompt.*

Note

When QinQ is disabled, all VLANs must be C-VLANs. When QinQ is enabled in svlan mode, all VLANs must be S-VLANs. When QinQ is enabled in mixed vlan mode, VLANs can be configured as either C-VLANs or S-VLANs. For more on S-VLAN configuration constraints, refer to the restrictions on page 8-10.

Configuring Per-Port S-VLAN Membership

The **svlan** <*vid*> command supports tagged and untagged options to configure per-port S-VLAN memberships. You can use these options from the configuration level by beginning the command with **svlan <** *vid* **>**, or from the context level of the specific VLAN by just typing the command option.

Syntax: *s*vlan < *vid* >

tagged < *port-list* >

Configures the indicated port(s) as **Tagged** *for the specified S-VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**.

untagged < *port-list* >

Configures the indicated port(s) as **Untagged** *for the specified S-VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**.

forbid < *port-list* >

QinQ svlan mode only. Used in port-based S-VLANs to configure **<** *port-list* **>** *as "forbidden" to become a member of the specified VLAN, as well as other actions. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto**. *Refer to chapter 3, "GVRP", in this guide.*

auto < *port-list* >

QinQ svlan mode only. Available if GVRP is enabled on the switch. Returns the per-port settings for the specified S-VLAN to **Auto** *operation. Note that* **Auto** *is the default per-port setting for a static VLAN if GVRP is running on the switch. Refer to chapter 3, "GVRP", in this guide.*

Note: *Since provider-gvrp is not supported in a QinQ mixed vlan mode environment, the* ***forbid*** *and* ***auto*** *configurations are only available in QinQ svlan mode. For more information on dynamic VLAN and GVRP operation, refer to chapter 3, "GVRP" in this guide.*

In QinQ mixed vlan mode. An interface (port or trunk) must be explicitly GVRP-disabled before it can be assigned to the S-VLAN space. When you first attempt to configure a port as tagged for an S-VLAN, the CLI will issue a message disallowing the configuration. For example:

```
<config #> svlan 200 tagged a1,a2
GVRP enabled ports cannot be members of svlans. Disable
the interface level gvrp configuration.
```

To disable gvrp at the interface, you would issue the following command:

```
<config #> interface a1,a2 unknown-vlans disable
```

Now when you configure the port, the CLI will issue a warning prompt:

```
<config #> svlan 200 tagged a1,a2
Ports a1, a2 will lose their cvlan memberships if any.
Do you want to continue? [y/n] y
```

Press [**Y**] to continue and automatically configure both ports as port-type 'provider-network' (the default for all S-VLAN member ports).

Configuring Port-Types

When QinQ is enabled on the switch all S-VLAN member ports must be categorized as either port-type customer-network or provider-network (see Figure 8-5).

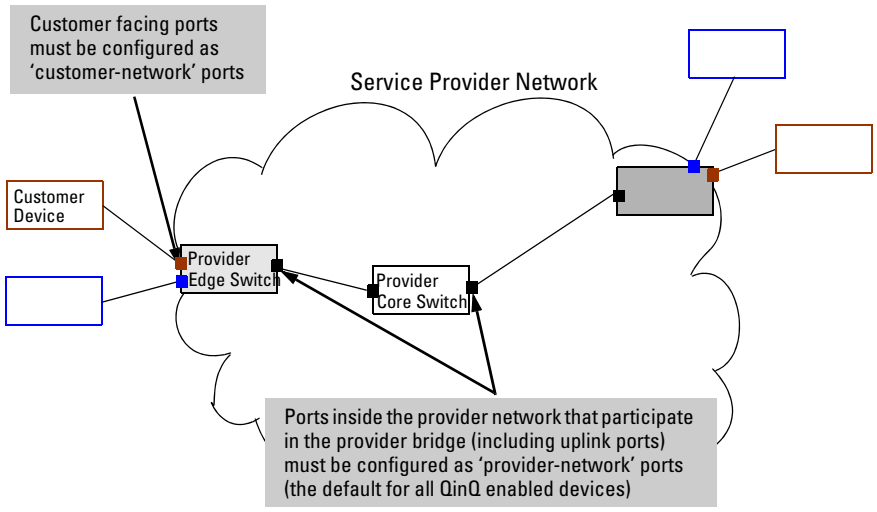

Customer facing ports must be configured as 'customer-network' ports
Service Provider Network
Customer Device
Provider Edge Switch
Provider Core Switch
Ports inside the provider network that participate in the provider bridge (including uplink ports) must be configured as 'provider-network' ports (the default for all QinQ enabled devices)

Figure 8-5. Example of Customer or Provider Ports in the Provider Network

All ports of a QinQ enabled device default to 'provider-network'. Any ports participating in the provider bridge that are used to connect to customer equipment, must be manually configured as port-type 'customer-network'. In a mixed mode device, ports that are members of C-VLANs and that do not participate in the provider-bridge cannot be configured to any port-type.

The following command allows you to configure the appropriate port-type.

Syntax: [no] interface <*port-list* | *Trkx*> qinq port-type <*customer-network* | *provider-network*>

Configures the specified ports/trunks as a customer network port or provider network port.
Default: port-type provider (for QinQ svlan mode)

Configuration Example

Figure 8-6 shows a configuration example that uses four ProCurve switches to establish a QinQ tunnel through the provider network.

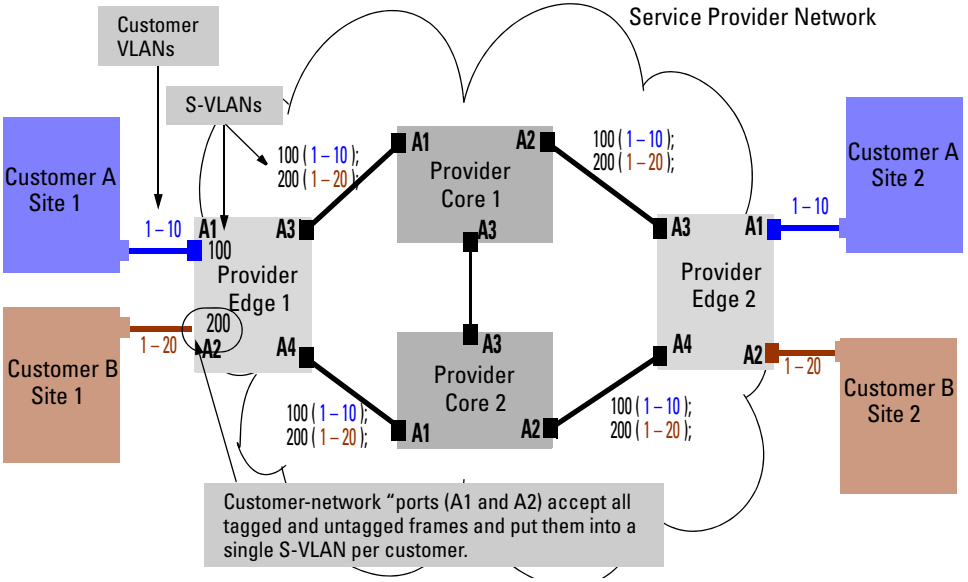


Figure 8-6. QinQ Configuration Example

The design parameters for this example are as follows:

- The provider edge bridge and the provider core bridge are configured in svlan mode.
- Each customer is associated with a single S-VLAN connecting two separate sites: customer A's VLANs (C-VLANs 1-10) are associated with S-VLAN 100; and customer B's VLANs (C-VLANs 1-20) are associated with S-VLAN 200.

Notes

- The VLANs of customers A and B can overlap: this will not result in intermixing of customer frames in the provider cloud because the S-VLANs associated with each customer are different.
- Core devices are not mandatory to establish a QinQ tunnel. For example, two edge-bridges can be connected directly to create a provider bridge network.

- The relationship between S-VLANs and C-VIDs is typically one to many. An alternative configuration might associate a single customer's C-VIDs with more than one S-VLAN. Such a configuration would most likely be used to tunnel distinct C-VIDs through various S-VLANs, but seldom be used to send the same C-VID through multiple S-VLANs.

Configure Provider Edge 1 Switch. Figure 8-7 shows the configuration details for "Edge 1" switch.

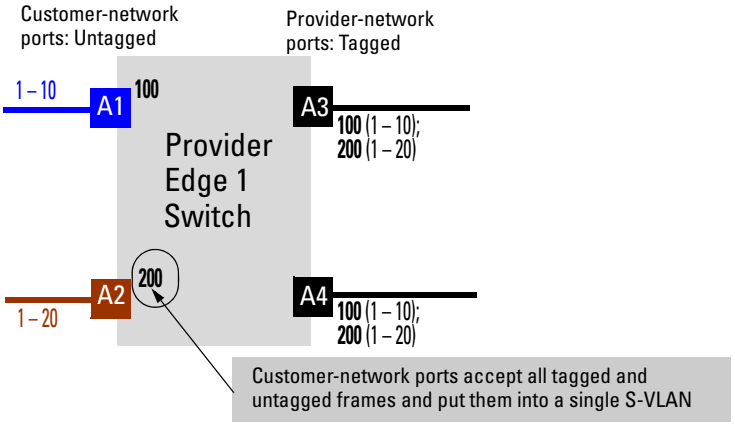


Figure 8-7. Configuration Example: Edge Switch 1

At the end of the configuration, the following settings will apply:

- All customer A site traffic received on port A1 will be associated with S-VLAN 100. This is independent of the C-VLAN tag information that the customer frames may carry.
- All customer B Site 1 traffic will be associated with S-VLAN 200 and be switched out to the core (uplinks A3, A4) with the S-VLAN tag-id of 200.
- The frame size will increase by 4 since ports A3 and A4 are tagged members of S-VLAN 100 and 200.

To configure the switch, you would do the following steps:

1. Enable QinQ.

   ```
   Edge l(config)# qinq svlan tag-type 88a8
   ```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.

Note

A reboot is required for the QinQ enable command to take effect.

3. Configure S-VLANs and ports connected to the customer network.

```
Edge1 (config)# svlan100
Edge1(svlan-100)# untagged A1
Edge1(svlan-100)# exit
Edge1(config)# int A1 qinq port-type customer-network
Edge1(config)# svlan 200
Edge1(svlan-200)# untagged A2
Edge1(svlan-200)# exit
Edge1(config)# int A2 qinq port-type customer-network
```

Note

In this example, customer A is assigned S-VLAN 100 and customer B is assigned S-VLAN 200. However, the same customer can be associated with more that one SVLAN. Also, interfaces A1 and A2 are configured as customer network ports because they are linked to customer bridges.

4. Configure the provider ports leading to the core of the provider network.

```
Edge1(config)# svlan 100 tagged A3, A4
Edge1(config)# svlan 200 tagged A3, A4
Edge1(config)# interface A3,A4 qinq port-type
provider-network
```

Note

As recommended by IEEE 802.1ad specification, uplink ports should generally be configured as tagged ports for S-VLANs that are used to carry customer traffic. However, this is not a mandatory requirement on ProCurve switches—S-VLANs that are used for internal provider network use (not carrying customer traffic but for management of the provider network devices) can have untagged port memberships.

Configure Provider Edge 2 Switch. The configuration details for the "Edge 2" switch mirrors the configuration for the Edge 1 switch. All customer traffic received on port A1 from customer A's site 2 will be associated with S-VLAN 100. Similarly, all customer B's site 2 traffic will be associated with S-VLAN 200.

To configure the switch, you would do the following steps:

1. Enable QinQ.

```
Edge 2(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.
3. Configure S-VLANs and customer ports connected to the customer network.

```
Edge2(config)# svlan 100
Edge2(svlan-100)# untagged A1
Edge2(svlan-100)# exit
Edge2(config)# int A1 qinq port-type customer-network
Edge2(config)# svlan 200
Edge2(svlan-200)# untagged A2
Edge2(svlan-200)# exit
Edge2(config)# int A2 qinq port-type customer-network
```

4. Configure the provider ports leading to the core of the provider network.

```
Edge1(config)# svlan 100 tagged A3, A4
Edge1(config)# svlan 200 tagged A3, A4
Edge1(config)# interface A3,A4 qinq port-type
provider-network
```

Configure Provider Core 1 Switch. Figure 8-8 shows the configuration details for the "Core 1" switch..

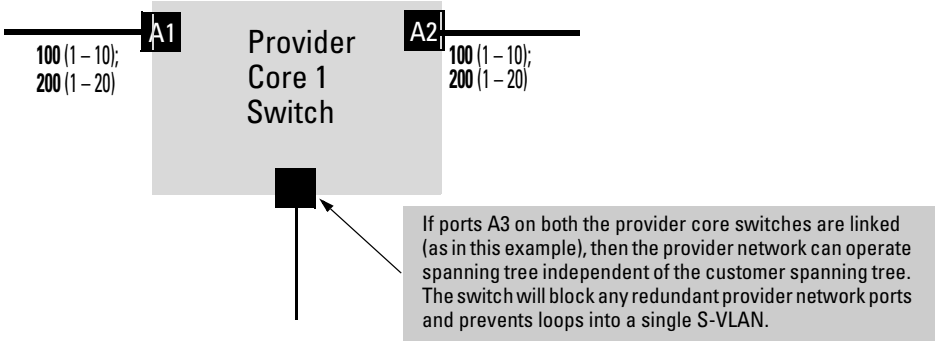

A1
100 (1 – 10);
200 (1 – 20)
Provider
Core 1
Switch
A2
100 (1 – 10);
200 (1 – 20)
If ports A3 on both the provider core switches are linked (as in this example), then the provider network can operate spanning tree independent of the customer spanning tree. The switch will block any redundant provider network ports and prevents loops into a single S-VLAN.

Figure 8-8. Configuration Example: Core Switch 1

To configure the Core 1 switch, you would take the following steps:

1. Enable QinQ.

```
Core l(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.
3. Configure S-VLANs and port assignments.

```
Core 1(config)# svlan 100
Core 1(svlan-100)# tagged A1, A2
Core 1(svlan-100)# exit
Core 1(config)# svlan 200
Core 1(svlan-200)# tagged A1, A2
Core 1(svlan-200)# exit
Core 1(config)# interface A1,A2 qinq port-type
provider-network
```

Note

The S-VLAN configuration for the core devices is based on what VLANs the edge devices (Edge 1 and 2) can send. Per the 802.1ad specification, all ports carrying customer traffic will be tagged on the VLAN that the port carries customer frames on.

Configure Provider Core 2 Switch. To configure the Core 2 switch, you would take the following steps:

1. Enable QinQ.

```
Core 2(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.
3. Configure S-VLANs and port assignments.

```
Core 2(config)# svlan 100
Core 2(svlan-100)# tagged A1, A2
Core 2(svlan-100)# exit
Core 2(config)# svlan 200
Core 2(svlan-100)# tagged A1, A2
Core 2(svlan-100)# exit
Core 2(config)# interface A1,A2 qinq port-type
provider-network
```

Verify the Configuration. Once the edge and core switch configurations are completed, QinQ operations can begin. To verify operations, it should be possible to assign IP-addresses to customer A or B devices in site 1 and site 2 and ping them. If everything has been configured correctly, traffic will flow through the provider network cloud and reach the other site seamlessly. To verify the configuration, see also “Displaying QinQ Config and Status” on page 8-24.

Updating QinQ Configurations

This section considers the impacts of updating QinQ modes and configuration settings on the switch.

Changing QinQ Modes

Changing QinQ modes (and/or disabling QinQ operations) will result in the current configuration being erased. See the following Caution for details.

Caution

Configuring the switch to operate in a different bridge mode requires a reboot to take effect. Upon reboot, **all configuration information for the prior QinQ mode will be lost**. Any configurations created under the existing QinQ mode will be erased, and the device will boot up with a default configuration for the new QinQ mode.

For information on the effect of the different QinQ modes on switch protocols and operations, refer to Table 8-2 on page 8-28.

Disabling QinQ

To disable QinQ once it has been enabled, you would issue the following commands from configuration mode on the CLI.

Syntax: no qinq

This is the default mode when QinQ is disabled on the switch.

Moving into this configuration from another qinq configuration requires a reboot to take effect. Upon reboot, all configuration information for the prior QinQ mode will be lost.

Default setting. Standard VLAN operations apply.

Changing VLAN Port Memberships (Mixed Vlan Mode)

On mixed vlan mode devices, certain per-port features are not supported on S-VLANs that are supported on C-VLANs. Ports that are currently members of a regular VLAN can only move to an S-VLAN if there is no conflicting configuration.

Note To avoid a misconfiguration, it is recommended that you use a default interface configuration when moving ports between C-VLANs and S-VLANs.

When configuring S-VLAN port memberships using the **svlan** command, the CLI issues a warning and prompt if any of the ports listed already belong to a regular VLAN. For example:

```
<config #> svlan 200 tagged a1,a2
Ports a1, a2 will lose their cvlan memberships if any.
Do you want to continue: y/n?
```

The warning prompt is displayed only when there is at least one port in the port list that needs to be moved out from the C-VLAN space to the S-VLAN domain. Similarly, if ports being added to the C-VLAN are already members of an S-VLAN, the CLI issues a warning that the port's membership with its existing VLANs will be removed and will prompt for a confirmation before continuing.

If all ports are just being added or removed from within the same VLAN type domain, no prompt will appear. For example, moving ports from S-VLAN 200 to S-VLAN 300, will not result in any warning as the ports are already part of the S-VLAN domain.

Moving Ports between C-VLANs and S-VLANs (Mixed Vlan Mode)

A port (or trunk) that is a member of C-VLANs cannot be moved into the S-VLAN space with conflicting configurations for the S-VLAN mode. The list of conflicting protocols/features is listed below. If a port has any of these enabled, the feature must be disabled before the port can be moved in to the S-VLAN space.

- An interface has to be GVRP-disabled to move it from the C-VLAN to the S-VLAN space. This is because S-VLANs of mixed vlan mode do not support provider-GVRP, and also because a GVRP-enabled configuration (when the port is a C-VLAN member) is in the context of customer-GVRP which must be disabled before the port can operate in the S-VLAN space.
- Interface should not have any mirroring or monitoring sessions when moving between C-VLANs and S-VLANs. All mirror/monitor sessions that involve the port must be unconfigured.
- An interface that has auth-vid or unauth-vid configuration cannot move into the S-VLAN space. They have to be unset.
- Interfaces cannot have LACP enabled (active or passive modes) when moving into the S-VLAN space. They have be disabled.

Displaying QinQ Config and Status

The following section outlines changes and additions to existing show command outputs to display QinQ configuration and status.

Show Commands for QinQ

The **show qinq** command displays QinQ configuration information.

Syntax: show qinq

Shows QinQ global and port configurations on the switch.

Mode: *Shows the QinQ configuration mode that is operating on the switch.*

- ***cvlan bridge****: QinQ is disabled, normal VLANs apply.*
- ***mixedvlan bridge mode****: Both S-VLANs and regular C-VLANs are available in a mixed vlan mode environment.*
- ***svlan mode****: No regular VLAN commands are available. All VLANs configured on the switch are S-VLANs only.*

Tag-id: *Displayed only if QinQ is enabled on the switch.*

Port Type: *Displayed only if QinQ is enabled on the switch. On a mixed mode device, port type is only shown for svlan ports.*

For example:

```
ProCurve (config)# show qinq

QinQ Global Configuration:
------------------------------------------------
Bridge-mode         : svlan bridge

QinQ Interface Configuration:
------------------------------------------------
interface    port-type
---------    ----------
   A1        provider-network
   A2        provider-network
   Trk1      customer-network
```

Figure 8-9. Example of "show qinq" Output (QinQ svlan mode)

Show Commands for VLANs

The following show commands are a subset of those listed in the chapter on Static Virtual LANs (VLANs) highlighting the changes made to show the additional QinQ VLAN types (C-VLANs and S-VLANs). For a full listing of all command parameters, refer to the chapter on Static Virtual LANs (VLANs).

Displaying the Switch's VLAN Configuration. The **show vlans** command lists the VLANs currently running in the switch, including the VID, VLAN name, and VLAN status. Once QinQ is enabled in mixed vlan mode, an additional field showing the VLAN type is added to the display output.

Syntax: show vlans

(Changes to parameters when QinQ is enabled:)

VLAN ID: *Field name changes from 802.1Q VLAN ID to VLAN ID only.*

Type: *In a QinQ mixed mode environment, the VLAN type can be either a regular customer VLAN* ***CVLAN****, or it can be a tunnel VLAN in the provider network* ***SVLAN****.*

For example:

```
ProCurve (config)# show vlans
Status and Counters - VLAN Information

Maximum VLANs to support : 256
Primary VLAN : DEFAULT_VLAN
Management VLAN : VLAN-100

VLAN ID Name         Type    | Status     Voice Jumbo
------- ------------ ------  + ---------- ----- ----
1       DEFAULT_VLAN CVLAN   | Port-based  No    No
10      Vlan-10      SVLAN   | Port-based  No    No
100     Vlan-100     CVLAN   | Port-based  No    No
101     Vlan-101     SVLAN   | Port-based  No    No
```

When QinQ is disabled (the default), S-VLANs do not exist on the switch and the VLAN Type field does not appear.

Figure 8-10. Example of "show vlan" Output (QinQ Mixed VLAN Mode)

Displaying the Configuration for a Particular VLAN. This command uses the VID to identify and display the data for a specific VLAN. Once QinQ is enabled in mixed vlan mode, an additional field showing the VLAN Type is added to the display output.

Syntax: show vlans < *vlan-id* >

(Changes to parameters when QinQ is enabled:)

VLAN ID: *Field name changes from 802.1Q VLAN ID to VLAN ID only.*

Type: *In a QinQ enabled environment, the VLAN type can be either a regular customer VLAN* ***CVLAN****, or it can be a tunnel VLAN in the provider network* ***SVLAN****.*

For example:

```
ProCurve Switch (config)# show vlan 10

Status and Counters - VLAN Information - Ports - VLAN 10

  VLAN ID : 10
  Name    : Vlan-10
  Type    : SVLAN
  Status  : Port-based
  Voice   : No
  Jumbo   : No

Port Information  Mode       Unknown VLAN  Status
----------------  ---------  ------------  ------
 1                Untagged   Disable       Down
 2                Untagged   Disable       Down
 3                Untagged   Disable       Down
 4                Untagged   Disable       Down
 5                Untagged   Disable       Down
```

When QinQ is enabled, the VLAN Type field is displayed.

Figure 8-11. Example of "Show VLAN" for a Specific VLAN (QinQ Enabled)

Displaying the VLAN Membership of One or More Ports. This command shows to which VLAN a port belongs. Once QinQ is enabled, an additional field showing the VLAN Type is added to the display output.

Syntax: show vlans < *vlan-id* >

(Changes to parameters when QinQ is enabled:)

VLAN ID: *Field name changes from 802.1Q VLAN ID to VLAN ID only.*

Type: *In a QinQ enabled environment, the VLAN type can be either a regular customer VLAN* ***CVLAN****, or it can be a tunnel VLAN in the provider network* ***SVLAN****.*

For example:

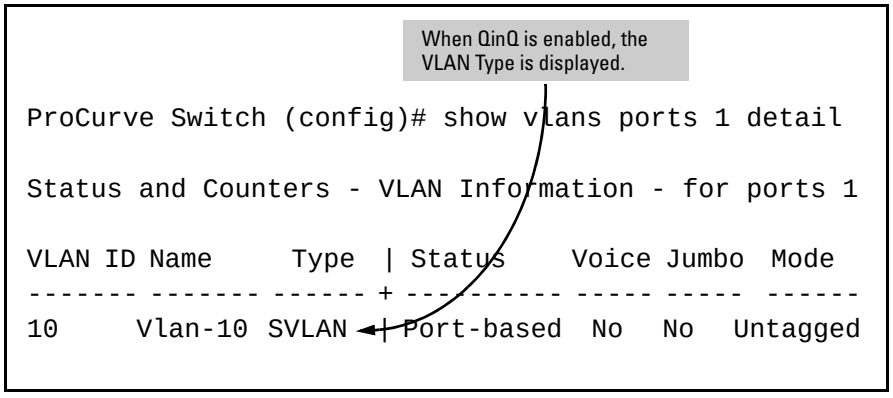

When QinQ is enabled, the VLAN Type is displayed.
ProCurve Switch (config)# show vlans ports 1 detail
Status and Counters - VLAN Information - for ports 1
VLAN ID Name Type | Status Voice Jumbo Mode
10 Vlan-10 SVLAN | Port-based No No Untagged

Figure 8-12. Example of "Show VLAN Ports" Output (QinQ Enabled)

Displaying Spanning Tree Status

In QinQ mixed mode, only ports that are members of C-VLANs will be displayed in **show spanning tree** output. This is due to the fact that ports that are members of S-VLANs do not participate in C-VLAN spanning tree and will always be in forwarding state (treated as edge ports).

Effects of QinQ on Other Switch Features

Per the IEEE standards, protocols such as STP and GVRP are assigned separate addresses for customer networks and provider networks, ensuring that QinQ has no impact on their operations. Bridge Protocol Data Units (BPDUs) that need to be tunneled through the provider network are treated as normal multicast frames at the provider bridge and forwarded out.

However, other protocols use common addresses for both customer and provider networks, and so are not supported when QinQ is enabled on the switch. Similarly, proprietary features such as meshing, discovery, UDLD, and loop-protect do not provide 'tunneling' support. In such cases, where provider networks could run an instance of the same protocol as a customer could run local to their site, these frames are dropped at the customer-network ports of the provider bridge.

Note

The IEEE standards group are devising new addressing schemes that may support additional QinQ tunneling operations. Check the latest product release notes for implementation updates as they apply to ProCurve switches.

When QinQ is not enabled (the default setting), there are no impacts to the switch's normal operations. Table 8-2 shows the impacts of QinQ on the operation of switch protocols and features based on the QinQ mode that is configured: QinQ mixed vlan mode (C-VLANs and S-VLANs are allowed) or QinQ svlan mode (S-VLANs only).

Table 8-2. Impacts of QinQ Configurations on Other Switch Features

Switch Feature	Impacts of QinQ Configurations and Allowed Operations
ACLs	**In QinQ mixed vlan or svlan modes**: • On double-tagged frames , the VID applicable when applying ACLs will be the S-VLAN tag and not the C-VLAN tag.
aaa	**In QinQ mixed vlan mode**: • auth-vid/unauth-vid configuration is not supported on svlan ports; the auth-vid/unauth-vid cannot be an S-VLAN id. • If a port that is a member of C-VLANs is configured with auth-vid or unauth-vid and it needs to be added to the S-VLAN domain, the auth/unauth configuration must first be undone.
arp-protect	**In QinQ mixed vlan mode**: • arp-protect is NOT supported on SVLANs, nor on S-VLAN ports.

CDP	**In QinQ mixed vlan or svlan modes**: • CDP frames are consumed at customer network ports, if CDP is enabled on the device port, and the customer device shows up as a CDP neighbor on the customer-network port. If not, the frames are dropped.
DHCP	**In QinQ mixed vlan or svlan modes**: • dhcp relay only applies to C-VLANs. • dhcp snooping is NOT supported on S-VLANs.
directed-broadcast	**In QinQ svlan mode**: • directed-broadcast is NOT supported on provider core devices.
GVRP	**In QinQ mixed vlan mode**: • S-VLAN ports cannot be GVRP enabled. • Regular VLANs will participate in C-VLAN GVRP if enabled to do so. S-VLANs will tunnel all C-VLAN GVRP frames through. • An explicit GVRP disable on a port is a prerequisite to moving the port to an S-VLAN domain. • Port-based interfaces do not have support for provider-GVRP protocols. Provider GVRP frames received at S-VLAN interfaces will be dropped. • If a VLAN being configured as an S-VLAN is already a GVRP VLAN on the switch, this S-VLAN creation would be blocked. **In QinQ svlan mode**: • GVRP is supported on S-VLAN ports if the qinq mode is svlan.
igmp-proxy	**In QinQ mixed vlan mode**: • igmp-proxy cannot be configured on S-VLANs. **In QinQ svlan mode**: • igmp-proxy is NOT supported.
IP	See "Layer 3 Protocols (IP, IP+, DHCP, ARP, IGMP Layer 3, Layer 3 ACLs)" on page 8-30.
IPv6	**In QinQ mixed vlan mode**: • IPv6 features are NOT supported on S-VLANs.
ip-recv-mac	**In QinQ mixed vlan mode**: • ip-recv-mac cannot be configured on S-VLANs. **In QinQ svlan mode**: • ip-recv-mac is NOT supported.
Jumbo	**In QinQ mixed vlan or svlan modes**: • No change in operations. It is recommended to jumbo-enable all SVLANs used for customer data tunneling to support the addition of the extra S-tag in each frame.

LACP/ Port Trunks	**In QinQ mixed vlan mode**: • Dynamic-LACP is not supported on S-VLAN ports: LACP manual trunks alone are supported. The new trunk will be a member of C-VLANs (port types are not applicable). • If two ports are added to a trunk, the resultant trunk will be a member of the default-vlan (vid-1) which is always a C-VLAN. The trunk can subsequently be manually assigned to an S-VLAN. • Port-type and VLAN configurations are not mapped. If the port-type is updated through CLI or SNMP and the port is subsequently moved from the C-VLAN space to the S-VLAN space then back again, the last configured port-type is retained through each move. **In QinQ svlan mode**: • On svlan bridges, both manual and dynamic LACP trunks are supported. It is not recommended to configure dynamic trunks on 'customer' ports because they cannot become dynamic members of SVLANs (there is no provider-gvrp for a dynamic trunk to become a member of S-VLANs.) • A newly formed trunk will by default be of type 'provider-network'. When the trunk is manually assigned to an S-VLAN for the first time after being created, the port-type is 'provider-network'.
Layer 3 Protocols (IP, IP+, DHCP, ARP, IGMP Layer 3, Layer 3 ACLs)	**In QinQ mixed vlan mode**: • There is no IP layer functionality on S-VLANs. • No change in IP layer functionality on regular C-VLANs. • S-VLANs cannot be configured as RIP, OSPF, PIM, or VRRP interfaces. **In QinQ svlan mode**: • S-VLANs can be ip enabled. • IP routing is NOT supported.
LLDP	**In QinQ mixed vlan or svlan modes**: • LLDP is supported on the device (in both qinq modes). However, there is no provision for tunneling customer LLDP BPDUs through the provider-network. • LLDP BPDUs received from a customer's network will be consumed at the customer-network ports of a provider device and the customer device will be displayed as an LLDP neighbor. Similarly the provider network device will show up as a neighbor on the customer's network if the customer-network ports send out LLDP advertisements.
load-sharing	**In QinQ svlan mode**: • Equal cost multi-path (ECMP) is NOT supported on provider core devices.
management VLAN	**In QinQ mixed vlan mode**: • The management VLAN cannot be an S-VLAN.
Meshing	**In QinQ mixed vlan mode**: • Meshing is NOT supported on the device. **In QinQ svlan mode**: • On an all provider-network ports of an svlan-bridge, meshing is supported. • Meshing cannot be enabled on customer-network ports.

Mirroring/ Monitoring	**In QinQ mixed vlan mode**: • Remote mirroring is not supported on S-VLANs. • Cannot monitor a VLAN with mirror ports in the other VLAN domain. That is, an S-VLAN or an S-VLAN port cannot be monitored using a C-VLAN port as its mirror, and vice-versa. • When a port is moved from the S-VLAN space to the C-VLAN space (or vice versa), all mirror/monitor sessions on the port must be unconfigured before the move will be allowed.
multicast-routing	**In QinQ svlan mode**: • Multicast routing is NOT supported on provider core devices.
QoS	**In QinQ mixed vlan or svlan modes**: • It is not recommended to enable DSCP on S-VLANs used for tunneling as the customer ip-pkt will be modified in the S-VLAN space.
Routing	**In QinQ svlan mode**: • Routing is NOT supported on provider core devices.
source-binding	**In QinQ mixed vlan or svlan modes**: • source-binding cannot be configured on S-VLANs.
source-route	**In QinQ svlan mode**: • source-route is NOT supported on provider core devices.
Spanning Tree	**In QinQ mixed vlan mode**: • Customer (C-VLAN) spanning tree is supported. All C-VLAN ports will receive/transmit customer STP BPDUs and participate in regular VLAN spanning tree as usual. • When customer STP BPDUs are received at S-VLAN ports on the switch, they will be flooded out of the other ports on the S-VLAN. All such frames will be tunneled through the S-VLAN tunnel unscathed. • Provider (S-VLAN) spanning tree is NOT supported on the switch. If S-VLAN STP frames are received on any S-VLAN enabled ports, they will be reforwarded out of the other ports on the S-VLAN. • STP configuration on S-VLAN ports is not supported. • If a port that is a member of C-VLANs is moved into being a member of S-VLANs, the port would, by default, tunnel customer STP BPDUs. • If a C-VLAN port has been configured with any non-default STP parameters (such as **admin-edge**, **auto-edge**, and **bpdu-protect**) and is then moved into an S-VLAN, the port will be put into a forwarding state regardless of the STP configurations done when the port was a member of the C-VLAN. • MSTP instances cannot include S-VLANs. **In QinQ svlan mode**: • Provider (S-VLAN) spanning tree is supported—both provider-network ports and customer-network ports will receive/transmit provider STP BPDUs. • Customer (VLAN) spanning tree tunneling is supported on S-VLAN interfaces—customer-network or provider-network ports will tunnel customer STP BPDUs through the appropriate S-VLAN.

Stacking	**In QinQ mixed vlan mode**: • Stacking is only supported on C-VLANs. The device does not advertise itself (using the stack discovery protocol) in the S-VLAN space. **In QinQ svlan mode**: • Stacking discovery protocol frames will not be sent out of customer-network ports; similarly, any stacking discovery protocol frames received on customer-network ports will be dropped.
UDLD	**In QinQ mixed vlan or svlan modes**: • UDLD frames received on udld-disabled customer network ports will be dropped. However, if the customer-network port is udld-enabled, it can peer with a customer device. • UDLD frames received on udld-disabled provider network ports will be reforwarded out of other udld-disabled provider network ports on the same VLAN. • UDLD reforwarding in the C-VLAN space (QinQ disabled or mixed vlan mode) will remain unaltered.
udp-bcast-forward	**In QinQ svlan mode**: • udp-bcast-forward is NOT supported on provider core devices.
unknown-vlans	**In QinQ mixed vlan mode**: • GVRP (learn and disabled modes) not supported on S-VLAN ports. • A C-VLAN port that has GVRP enabled will need to disable it before it can be added to S-VLANs.
Voice VLANs	**In QinQ mixed vlan mode**: • S-VLANs cannot be configured as voice-VLANs.
VRRP	**In QinQ mixed vlan or svlan modes**: • VRRP is NOT supported on S-VLANs.

Event Log Messages and SNMP Support

Table 8-3 shows the event log messages that may be generated when QinQ is enabled on the switch.

Table 8-3. QinQ Event Log Messages

Message	Meaning
`system: Rebooting for qinq mode change`	System reboot occasioned by a change in QinQ configuration mode.
`QinQ: qinq mode - mixedvlan`	The device is configured to operate in qinq mixed vlan mode.
`QinQ: qinq mode - svlan`	The device is configured to operate in qinq svlan mode.

SNMP Support and MIB Objects

The switch SNMP agent can make use of certain variables that are included in a Hewlett-Packard proprietary MIB (Management Information Base) file. Table 8-4 on page 8-34 shows the MIB objects that can be used to configure the Q-in-Q (provider bridge) mode, the provider bridge port-type, and the QinQ outer tag ethertype.

The MIB objects listed in the following table may be subject to change. The **walkmib** <*OBJECT-STR*> CLI command can be used to verify the availability of the listed MIB objects on the device.

To download the latest version of the MIB file:

1. Go to the ProCurve Networking Web site at: www.procurve.com
2. Click on **software updates**, then **MIBs**.

Table 8-4. MIB Objects for QinQ

MIB Object	Notes and Descriptions
Standard MIBs	
dot1qVlanStaticTable	VLANS (S-VLANs or C-VLANS) are created using the standard MIBs such as the dot1qVlanStaticTable defined by rfc2674_q MIB. To differentiate the VLAN as an S-VLAN or a C-VLAN, the dot1qVlanStaticTable is augmented by an HP proprietary MIB (hpicfVlanClassifierTable) that lets the user specify the VLAN type.
dot1qVlanStaticEntry	A port can be a member of C-VLANs or S-VLANs but not both. When changing port memberships using the SNMP dot1qVlanStaticEntry mib object, variable binding to add a port to the svlan and variable bindings to remove the port from all existing cvlans (that it is a member of) need to be sent in the same PDU (much like how the 'vlan <vlan-id> untagged <port>' CLI command works today). The same is applicable when moving a port from the S-VLAN space to the C-VLAN space.
HP Proprietary MIBs	
hpicfProviderBridge	This MIB module is the HP 'version' of the standard Provider Bridge MIB and the proprietary extensions to it.
hpicfProviderBridgeType	Configures a device to any one of the following bridge mode options: • *regular vlanBridge* (provider bridge feature disabled mode, all VLANs are C-VLANs); • *s-vlan bridge* (provider bridge mode with only S-VLANs); • *provider edge bridge* (provider bridge mode with C-VLANs and S-VLANs and mappings between them); • *vlanSvlanBridge* (provider bridge mode with independent C-VLANs and S-VLANs on the same device).
hpicfProviderBridgeEtherType	Defines the 2-byte ethertype for provider-tagged frames. It is applicable to the following bridge types: s-vlan bridge, provider edge bridge, or vlanSvlanBridge. The default value is 0x88a8. Changing from one tag-type to another with a given hpicfProviderBridgeType configuration will reboot the device and the new tag-type will take effect subsequently.
hpicfProviderBridgeVlanType Table	Augments the dot1qVlanStaticTable (see Standard MIBs above). Used to configure a VLAN as an S-VLAN or as a C-VLAN in vlanSvlanBridge mode. The absence of this variable binding would classify the new VLAN as a C-VLAN.

MIB Object	Notes and Descriptions
hpicfProviderBridgePortTable	Used to configure each port as a 'provider' or 'customer'. This MIB object is only relevant when the device is QinQ enabled. On a non provider device (QinQ disabled), any SET operation on this MIB object is disallowed and an 'SNMP-GET' would not return any entry.
hpicfProviderBridgePortEntry	Used to specify the designated type of an externally accessible port on a Provider Bridge. These include the following options: • *customer-edge port* (reserved for future use); • *customer-network port*: an S-VLAN component port on a provider bridge or within a provider edge bridge that receives and transmits frame for a single customer. • *provider-network port*: an S-VLAN component port on a provider bridge that can transmit and receive frames for multiple customers.

9

Classifier-Based Software Configuration

Contents

Using Classifier-Based Service Policies 9-2
- Introduction 9-2
- Classifier-Based Configuration Model 9-2

Creating a Traffic Class 9-4
- Using Match Criteria 9-4
- Class Configuration Procedure 9-5
- Optional ICMP Match Criteria 9-13
- Optional IGMP Match Criteria 9-16
- Optional TCP and UDP Match Criteria 9-17
- Using CIDR Notation for IPv4/IPv6 Addresses 9-19
- Resequencing Match/Ignore Statements 9-23

Creating a Service Policy 9-24
- Modifying Classes in a Policy 9-28
- Resequencing Classes in a Policy 9-29

Applying a Service Policy to an Interface 9-30

Where to Go From Here 9-33

Using Classifier-Based Service Policies

Introduction

Classifier-based service policies are designed to work with existing globally-configured, switch-wide and port-wide configurations by allowing you to zoom in on a subset of port or VLAN traffic to further manage it. These policies take precedence over, and may override, globally-configured settings.

Classifier-based service policies provide greater control for managing network traffic. Using multiple match criteria, you can finely select and define the classes of traffic that you want to manage. Policy actions determine how you can handle the selected traffic.

Starting in software release K.14.01, the Classifier feature introduces:

- A finer granularity than globally-configured features for classifying network traffic (IPv4 or IPv6) into classes that can be used in cross-feature software configurations
- Additional policy actions to manage selected traffic, such as rate-limiting and IP precedence marking
- The configuration of service policies for classified traffic with the following software features:
  - Quality of Service (QoS)
  - Traffic mirroring
- The application of service policies to specific inbound traffic flows on individual port and VLAN interfaces (rather than only on switch-wide or port-wide traffic).

Classifier-Based Configuration Model

Classifier-Based Configuration Task	Page Reference
Creating a Traffic Class	page 9-4
Creating a Service Policy	page 9-24
Applying a a Service Policy to an Interface	page 9-30

Classifier-based software configuration consists of the following general steps:

1. Determine the inbound traffic you want to manage and how you want to manage it; for example, rate-limit, prioritize, mirror, and so on.
2. Classify the traffic that you want to manage by configuring a class, using **match** and **ignore** commands. A traffic class is configured separately from service policies and can be used in various policies.
3. Configure a service policy for one or more traffic classes, including an optional, default class. A policy consists of configuration commands executed on specified traffic classes for one of the following software features:
   - Quality of Service (**policy qos** command)
   - Port mirroring (**policy mirror** command)
4. Assign the policy to an inbound port or VLAN interface using the **interface service-policy in** or **vlan service-policy in** command.

Figure 9-1 shows an overview of classifier-based software configuration.

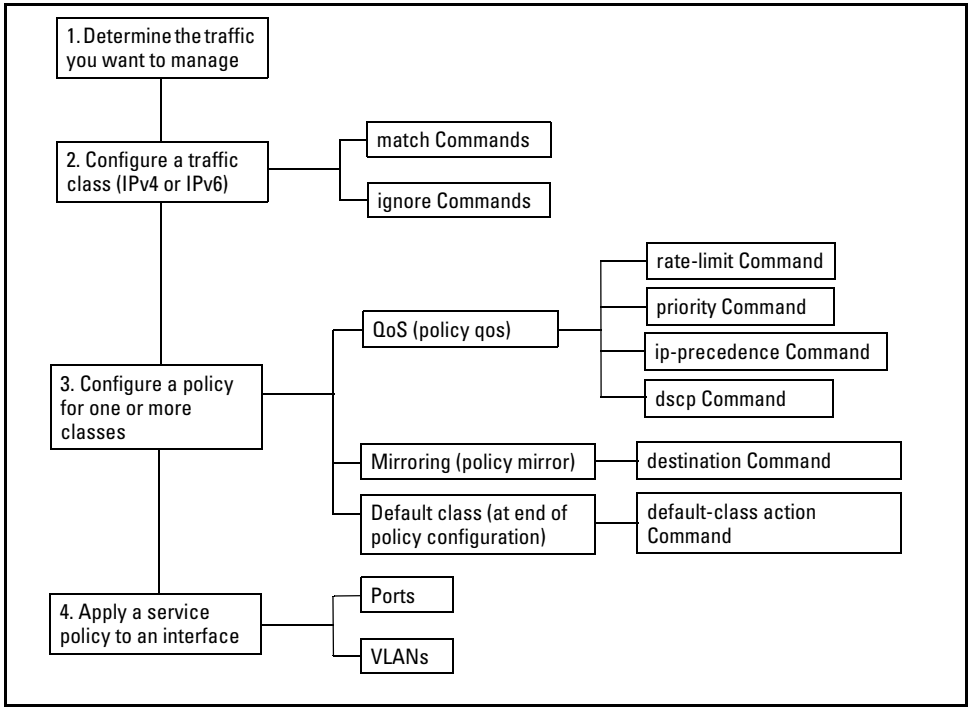


Figure 9-1. Classifier-Based Configuration Model

Creating a Traffic Class

In the classifier-based configuration model, you use match criteria to create a class of IPv4 or IPv6 traffic and select the packets you want to manage. In a class configuration, match criteria consist of **match** and **ignore** commands. These commands determine the packets that belong to a class. (Match/ignore criteria are modelled on the permit/deny criteria used in ACLs.)

The traffic classes you configure can be used later in the service policies you create for different software features, such as QoS and port mirroring. The match criteria used in match/ignore statements are the same across software features.

Using Match Criteria

To identify the packets that belong to a traffic class for further processing by policy actions, use **match** and **ignore** commands in a class configuration:

- **match** commands define the values that header fields must contain for a packet to belong to the class and be managed by policy actions.
- **ignore** commands define the values which, if contained in header fields, exclude a packet from the policy actions configured for the class. An ignored packet is transmitted without having a policy action performed on it.

Match/ignore statements compare the values in packet fields with specified criteria in the sequential order in which the statements are entered in the class, until a match is found. Be sure to enter match/ignore statements in the *precise order* in which you want their criteria to be used to check packets.

- As soon as a field in a packet header matches the criteria in a **match** statement, the sequential comparison of match criteria in the class stops, and the policy actions configured for the class are executed on the packet (see “Creating a Service Policy” on page 9-24).
- If a packet matches the criteria in an **ignore** statement, the sequential comparison of match criteria in the class stops, and no policy action is performed on the packet.

If a packet does not match the criteria in any match/ignore statement in a class configuration, one of the following actions is taken:

- The packet is transmitted without a policy action performed on it.

- If a default class is configured in the policy, the actions specified in the **default-class** command are performed on packets that do not match the criteria in preceding classes in the policy (see Step 3 in "Creating a Service Policy" on page 9-24).

The following match criteria are supported in match/ignore statements for inbound IPv4/IPv6 traffic:

- IP source address (IPv4 and IPv6)
- IP destination address (IPv4 and IPv6)
- Layer 2 802.1Q VLAN ID
- Layer 3 IP protocol
- Layer 3 IP precedence bits
- Layer 3 DSCP bits
- Layer 4 TCP/UDP application port (including TCP flags)
- VLAN ID

Class Configuration Procedure

To configure a traffic class to be used in one or more policies, follow these steps:

1. Enter the **class** command from the global configuration context.

Context: Global configuration

Syntax: [no] class < ipv4 | ipv6 > <*classname* >

Defines a traffic class and specifies whether a policy is to be applied to IPv4 or IPv6 packets, where < classname > is a text string (64 characters maximum).

After you enter the **class** *command, you enter the class configuration context to specify match criteria. A traffic class contains a series of* **match** *and* **ignore** *commands, which specify the criteria used to classify packets.*

2. Enter one or more **match** or **ignore** commands from the class configuration context to filter traffic and determine the packets on which policy actions will be performed.

Context: Class configuration

Syntax: [no] [*seq-number*] < match | ignore > < *ip-protocol* > < *source-address* > < *destination-address* > [ip-dscp *codepoint*] [precedence *precedence-value*] [tos *tos-value*] [vlan *vlan-id*]

[*seq-number*]

The (optional) ***seq-number*** *parameter sequentially orders the match/ignore statements that you enter in a traffic class configuration. Packets are checked by the statements in numerical order.*

Default: Match/ignore statements are numbered in increments of 10, starting at 10. To re-number the match/ignore statements in a class configuration, use the **resequence** *command (see "Resequencing Match/ Ignore Statements" on page 9-23).*

< match | ignore >

Defines the classifier criteria used to determine which packets belong to the traffic class.

If a packet matches a **match** *criterion, it becomes a member of the traffic class and is forwarded according to the actions configured with the* **policy** *command. If a packet matches an* **ignore** *criterion, no policy action is performed on the packet. You can enter one or more match/ignore statements in a traffic class.*

To remove a match/ignore statement from a class configuration, enter the **no** < ***seq-number***> *command or the complete form of a* **no match ...** *or* **no ignore ...** *command.*

< *ip-protocol* >

Specifies an IP protocol to be matched in packet fields of IPv4 or IPv6 traffic, where ***ip-protocol*** *is one of the values described below.*

When entering a match/ignore command in an IPv4 or IPv6 class, type **?** *to display a list of valid* ***ip-protocol*** *entries.*

- *In an IPv4 class, you can enter any of the following IPv4 protocol match criteria:*

ah	**esp**	**gre**	**icmp***	**igmp***
ip	**ip-in-ip**	**ipv6-in-ip**	**ospf**	**pim**
sctp	**snmp**	**tcp***	**udp***	**vrrp**

  ** For IPv4 ICMP, IGMP, TCP, and UDP packets, you can enter additional match criteria; see:*
  "Optional ICMP Match Criteria" on page 9-13
  "Optional IGMP Match Criteria" on page 9-16
  "Optional TCP and UDP Match Criteria" on page 9-17

  To specify an IPv4 protocol as match criteria, you can also enter its protocol number. Valid values are from **0** *to* **255**.

  For example, **8** *means Exterior Gateway Protocol;* **121** *means Simple Message Protocol. For a list of IPv4 protocol numbers and corresponding protocol names, refer to the IANA "Protocol Number Assignment Services" at www.iana.com.*

- *In an IPv6 class, you can enter any of the following IPv6 protocol match criteria:*

ah	**esp**	**icmp***	**ipv6**
sctp	**tcp***	**udp***	

  ** For IPv6 ICMP, TCP, and UDP packets, you can enter additional match criteria; see:*
  "Optional ICMP Match Criteria" on page 9-13
  "Optional TCP and UDP Match Criteria" on page 9-17

< *source-address* > < *destination-address* >

Define the source IP address (SA) and destination IP address (DA) that a packet must contain to match a match/ignore statement in an IPv4 or IPv6 traffic class. Note that both the source and destination address parameters are required entries in a match/ignore statement.

Valid values for < **source-address** > *and* < **destination-address** > *are as follows:*

- **any** — *Matches IPv4 or IPv6 packets from, or destined to, any SA or DA.*

- **host < *SA* | *DA* >** — *Matches only packets from a specified IPv4 or IPv6 host address. Use this match criterion when you want to match IP packets from only one SA/DA.*

- ***SAv4 mask* | *DAv4 mask*** — *Matches packets received from, or destined to, a subnet or a group of IP4 addresses defined by the IPv4 mask. Enter an IPv4 mask in dotted-decimal format for an IPv4 address (for example, 10.28.31.1 0.0.0.255).*

  See "Using CIDR Notation for IPv4/IPv6 Addresses" on page 9-19 for information on how IPv4 mask bit sets define a match.

  Note that an IPv6 address and mask are not supported as <SAv6 mask> and <DAv6 mask> match criteria.)

- ***SAv4/mask-length* | *DAv4/mask-length*** — *Matches packets received from, or destined to, an IPv4 subnet or a group of IPv4 addresses defined by the mask length. Enter the mask length for an IPv4 SA or DA mask in CIDR format by using the number of significant bits. (for example, 10.28.31.3/24).*

  An IPv4 mask-length is applied to an SA or DA in a match/ignore statement to define which bits in a packet's SA/DA must exactly match the specified SA/DA and which bits need not match.

  For example, 10.28.31.3/24 means that the leftmost 24 bits in an IPv4 source or destination address in a packet header must match the same bit set in the specified IPv4 address (in this case, 10.28.3.3).

  For more information, see "Using CIDR Notation for IPv4/IPv6 Addresses" on page 9-19.

  An IPv4 mask-length is applied from right to left, starting from the rightmost bits.

  ***Example:** 10.10.10.1/24 and 10.10.10.1 0.0.0.255 both match IPv4 addresses in the range 10.10.10.(1 to 255).*

  ***Note:** Specifying a group of non-contiguous IP source addresses may require more than one match/ignore statement.*

- ***SAv6/prefix-length | DAv6/prefix-length*** *— Matches packets received from, or destined to, an IPv6 subnet or a group of IPv6 addresses defined by the prefix length. Enter the prefix length for an IPv6 SA/DA in CIDR format by using the number of significant bits; for example: 2001:db8:2620:212::01b4/64.*

  An IPv6 prefix-length is applied to an SA/DA in a match/ignore statement to define which bits in a packet's SA/DA must exactly match the specified SA/DA and which bits need not match.

  For example, 2001:db8:2620:212::01b4/64 means that the leftmost 64 bits in a 128-bit IPv6 source or destination address in a packet header must match the same bit set in the specified IPv6 address (in this case, 2001:db8:2620:212::01b4).

  For more information, see "Using CIDR Notation for IPv4/IPv6 Addresses" on page 9-19.

  An IPv6 prefix-length is applied from left to right, starting from the leftmost bits.

  Example: *2001:db8::0001:2620:a03:e102:127/64 and 2001:db8::1:244:17ff:feb6:d37d/64 both match IPv6 addresses with a network prefix of 2001:db8:0000:0001.*

[ip-dscp *codepoint*]

(Optional) Matches the six-bit DSCP codepoint in IPv4 or IPv6 packets to further define match criteria. Valid values for ***codepoint*** *are one of the following:*
- Numeric equivalent of a binary DSCP bit set from **0** *(low priority) to* **63** *(high priority)*
-ASCII standard name for a binary DSCP bit set:

af11 (001010)	**af42** (100100)
af12 (001100)	**af43** (100110)
af13 (001110)	**ef** (101110)
af21 (010010)	**cs1** (001000) = *precedence 1*
af22 (010100)	**cs2** (010000) = *precedence 2*
af23 (010110)	**cs3** (011000) = *precedence 3*
af31 (011010)	**cs4** (100000) = *precedence 4*
af32 (011100)	**cs5** (101000) = *precedence 5*
af33 (011110)	**cs6** (110000) = *precedence 6*
af41 (100010)	**cs7** (111000) = *precedence 7*
default (000000)	

To display a list of valid ***codepoint*** *entries when you enter* **ip-dscp** *in a match/ignore statement, type* **?**. *The DSCP codepoints are the leftmost six bits of the ToS/Traffic Class byte (see Figure 9-2).*

[precedence *precedence-value*]

(Optional) Matches the three-bit IP precedence value in IPv4 or IPv6 packets to further define match criteria. Valid values for ***precedence-value*** *are either the numeric value (0 to 7) or corresponding name of an IP precedence bit set:*

0 routine
1 priority
2 immediate
3 flash
4 flash-override
5 critical
6 internet (for internetwork control)
7 network (for network control)

To display a list of valid ***precedence-value*** *entries when you enter* **precedence** *in a match/ignore statement, type* **?**.

Notes: *When used as a match criteria, the IP precedence value is applied in addition to all other criteria configured in the match/ignore statement. You can enter a match/ignore statement either with or without a* ***precedence-value****.*

The IP precedence bits are the leftmost three bits of the ToS/Traffic Class byte (see Figure 9-2). The numeric value (0 to 7) of the IP precedence bits corresponds to the hexadecimal equivalent of the three binary "0" and/or "1" bits in the IP precedence field. For example if the IP precedence-bit binary values are **1 1 1**, *the numeric value is* **7** *(1+2+4). Similarly, if the IP precedence bits are* **0 1 0**, *the numeric value is* **2** *(0+2+0).*

[tos *tos-value*]

(Optional) Matches the Delay Throughput Reliability (DTR) bit set in the IPv4 Type-of-Service or IPv6 Traffic Class byte to further define match criteria.

Valid values are the numeric value or corresponding name of the DTR bit set. Some useful values are as follows:

0 normal
2 max-reliability
4 max-throughput
8 minimize-delay

Default: **0** *or* **normal**.

To display a list of valid ***tos-value*** *entries when you enter* **tos** *in a match/ignore statement, type* **?**.

Notes: *When used as a match criteria, the ToS/Traffic Class byte entry is applied in addition to all other criteria configured in the match/ignore statement. You can enter a match/ignore statement either with or without a* ***tos-value***.

Figure 9-2 shows the DTR bit set in a ToS/Traffic Class byte in an IPv4/IPv6 packet header and the difference between the DSCP, DTR, and precedence bits.

[vlan *vlan-id*]

(Optional) Matches the VLAN ID number in the Layer 2 header of 802.1Q VLAN packets to further define match criteria. Valid VLAN IDs are from 1 to 4094.

Figure 9-2 uses a sample ToS/Traffic Class field of **10101000** *to show the differences between the IP precedence (***101***), DSCP (***101010***), and ToS/Traffic Class (***10101000***) bits. Note that the rightmost two bits are reserved as* **00**.

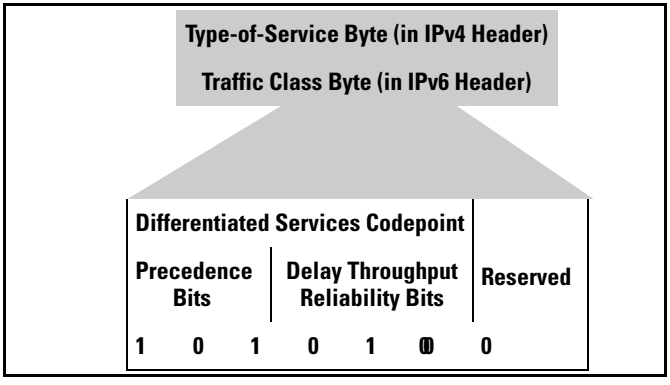

Type-of-Service Byte (in IPv4 Header)
Traffic Class Byte (in IPv6 Header)
Differentiated Services Codepoint
Precedence Bits
Delay Throughput Reliability Bits
Reserved
1 0 1 0 1 0 0 0

Figure 9-2. Example of a ToS/Traffic Class Field

3. Enter the **exit** command to exit the class configuration context.

   To display a class configuration, enter the **show class < ipv4 | ipv6 > < *classname* >** command (see Figure 9-7).

   To edit a class configuration, re-enter the class configuration context (**class** command) and enter new match/ignore statements as follows:

   - If you do not enter a sequence number, a new statement is inserted at the end of the class configuration.
   - To remove a match/ignore statement from a class configuration, enter the **no** <***sequence-number***> command or the complete form of the **no match ...** or **no ignore ...** command.
   - To resequence the order in which match/ignore statements are listed, enter the **resequence** command (see "Resequencing Match/Ignore Statements" on page 9-23).
   - To replace an existing match/ignore statement, enter the **no** <***sequence-number***> command to delete the entry and re-enter a complete <***sequence-number***> **match ...** or <***sequence-number***> **ignore ...** command.

   When you exit class configuration context, the changes are automatically saved and applied to existing policy configurations on the switch that use the class *if the policies have not been applied to an interface*. If a policy has already been applied to an interface, the editing changes are not accepted and an error message is displayed.

Example. Figure 9-3 shows an example of two class configurations:

- "AdminTraffic" selects the administrative traffic sent to, and received from, the IPv4 address of an administrator's PC.
- "http" selects HTTP traffic sent to TCP ports 80, 443, and 8080, and excludes HTTP traffic sent to, and received from, TCP port 1214.

```
ProCurve(config)# class ipv4 AdminTraffic
ProCurve(class-config)# match ip 15.29.16.1/10 any
ProCurve(class-config)# match ip any 15.29.16.1/10
ProCurve(class-config)# exit
ProCurve(config)# class ipv4 http
ProCurve(class-config)# match tcp any any eq 80
ProCurve(class-config)# match tcp any any eq 443
ProCurve(class-config)# match tcp any any eq 8080
ProCurve(class-config)# ignore tcp any eq 1214 any
ProCurve(class-config)# ignore tcp any any eq 1214
ProCurve(class-config)# exit
```

Figure 9-3. Example of a Class Configuration

Optional ICMP Match Criteria

To more precisely define the ICMP packets that you want to match in an IPv4 or IPv6 traffic class, use the optional parameter settings described in this section. For example, instead of matching or ignoring all ICMP traffic, you can configure a class that matches only a specific ICMP packet type by entering its numeric value.

Context: Class configuration

Syntax: [no] [*seq-number*] < match | ignore > **icmp**
< *source-address* > < *destination-address* >
[***icmp-type-number*** | ***icmpv4-type-name*** | ***icmpv6-type-name***]
[ip-dscp *codepoint*] [precedence *precedence-value*]
[tos *tos-value*] [vlan *vlan-id*]

If you enter **icmp** *as the IP protocol type in a match/ignore statement, you can optionally specify an ICMP packet type to more precisely define match criteria for a traffic class. Enter the optional ICMP match criteria immediately after the destination address (DA) value in the command syntax; for example:*

```
ProCurve(config-class)# match icmp any any host-
unknown
ProCurve(config-class)# match icmp any any 3 7
```

[*icmp-type-number*]

Configures an ICMP packet type as match criteria in a class configuration by entering its numeric identifier. Valid values are from **0** *to* **255**.

For information on ICMP packet-type names and numeric identifiers, go to the Internet Assigned Numbers Authority (IANA) website at **www.iana.com,** *click on "Protocol Number Assignment Services", and then go to the selections under "Internet Control Message Protocol (ICMP) Parameters".*

[*icmpv4-type-name*]

You can also enter any of the following ICMPv4 packet-type names to configure more precise match criteria for ICMP packets in an IPv4 class configuration.

To display a list of valid ***icmpv4-type-name*** *entries when you enter* **icmp** *as the IP protocol type in a match/ignore statement, type* **?**. *Some of the valid values are as follows:*

administratively-prohibited	net-tos-unreachable
alternate-address	net-unreachable
conversion-error	network-unknown
dod-host-prohibited	no-room-for-option
dod-net-prohibited	option-missing
echo	packet-too-big
echo-reply	parameter-problem
general-parameter-problem	port-unreachable
host-isolated	precedence-unreachable
host-precedence-unreachable	protocol-unreachable
host-redirect	reassembly-timeout
host-tos-redirect	redirect
host-tos-unreachable	router-advertisement
host-unknown	router-solicitation
host-unreachable	source-quench
information-reply	source-route-failed
information-request	time-exceeded
mask-reply	timestamp-reply
mask-request	timestamp-request
mobile-redirect	traceroute
net-redirect	ttl-exceeded
net-tos-redirect	unreachable

[*icmpv6-type-name*]

You can also enter any of the following ICMPv6 packet-type names to configure more precise match criteria for ICMP packets in an IPv6 class configuration.

To display a list of valid ***icmpv6-type-name*** *entries when you enter* **icmp** *as the IP protocol type in a match/ignore statement, type* **?**. *Some of the valid values are as follows:*

cert-path-advertise	mobile-advertise
cert-path-solicit	mobile-solicit
destination-unreachable	nd-na
echo-reply	nd-ns
echo-request	node-info
home-agent-reply	node-query
home-agent-request	packet-too-big
inv-nd-na	parameter-problem
inv-nd-ns	redirect
mcast-router-advertise	router-advertisement
mcast-router-solicit	router-renum
mcast-router-terminate	router-solicitation
mld-done	time-exceeded
mld-query	ver2-mld-report
mld-report	

Optional IGMP Match Criteria

To more precisely define the IGMP packets that you want to match in an IPv4 traffic class, use the optional parameter settings described in this section. For example, instead of matching all IGMP traffic, you can configure a class that matches only a specific IGMP packet type.

Context: Class configuration

Syntax: [no] [*seq-number*] < match | ignore > **igmp**
< *source-address* > < *destination-address* > [***igmp-type***]
[ip-dscp *codepoint*] [precedence *precedence-value*]
[tos *tos-value*] [vlan *vlan-id*]

If you enter **igmp** *as the IP protocol type in a match/ignore statement, you can optionally specify an IGMP packet type to more precisely define match criteria for a traffic class. Enter the optional IGMP match criteria immediately after the destination IP address (DA) value in the command syntax; for example:*

```
ProCurve(config-class)# match igmp any any host-
query
```

[*igmp-type*]

Configures an IGMP packet type as match criteria in a class configuration. Some of the valid values for IGMP packet-type names are as follows:

dvmrp	mtrace-request	trace
host-query	mtrace-reply	v2-host-leave
host-report	pim	v2-host-report
		v3-host-report

To display a list of valid ***igmp-type*** *entries when you enter* **igmp** *as the IP protocol type in a match/ignore statement, type* **?**.

Optional TCP and UDP Match Criteria

In a class configuration, you can enter match/ignore statements that more precisely define the TCP or UDP traffic that you want to match in an IPv4 or IPv6 traffic class. For example, you can enter a port number as a match criterion that specifies one or more TCP source ports, destination ports, or both.

Context: Class configuration

Syntax: [no] [*seq-number*] < match | ignore > < **tcp** | **udp** >
< *source-address* > [***operator*** < ***tcp-src-port*** | ***udp-src-port*** >]
< *destination-address* > [***operator*** < ***tcp-dest-port*** [**established**]
[***tcp-flag*** [***tcp-flag ...***]] | ***udp-dest-port*** >] [ip-dscp *codepoint*]
[precedence *precedence-value*] [tos *tos-value*] [vlan *vlan-id*]

If you use TCP or UDP as the IP protocol type in a match/ignore statement, you can optionally configure TCP or UDP source and/or destination port numbers or ranges of numbers to more precisely define match criteria for a traffic class. Enter the optional TCP/UDP match criteria immediately after the source and/or destination address in the command syntax; for example:

```
ProCurve(config-class)# match tcp host 10.20.10.17
eq 23 host 10.20.10.155 established
ProCurve(config-class)# match tcp host 10.10.10.100
host 10.20.10.17 eq telnet
ProCurve(config-class)# ignore udp 10.30.10.1/24 host
10.20.10.17 range 161 162
```

[*operator* < *tcp-src-port* | *udp-src-port* >]

To specify a TCP or UDP source port number as a match criteria, enter a comparison operator from the following list with a TDP/UDP port number or well-known port name immediately after the source-address value in the command.

Comparison Operators:

- **eq** < ***tcp/udp-port-number*** > — *"Equal To" matches a packet with the same TCP or UDP source port number as* < ***tcp/udp-port-number*** >.
- **gt** < ***tcp/udp-port-number*** > — *"Greater Than" matches any packet with a TCP or UDP source port number greater than* < ***tcp/udp-port-number*** >.
- **lt** < ***tcp/udp-port-number*** > — *"Less Than" matches any packet with a TCP or UDP source port number less than* < ***tcp/udp-port-number*** >.

- **neq** < ***tcp/udp-port-number***> — *"Not Equal" matches any packet with a TCP or UDP source port number that is not equal to* ***< tcp/udp-port-number >***.
- **range** ***< start-port-number > < end-port-number >*** — *Matches any packet with a TCP or UDP source port number in the range* <***start-port-number >*** *to* ***< end-port-number >***.

TCP/UDP Well-Known Source-Port Names and Numbers *— Enter a comparison operator with the source TCP or UDP port number used by the applications you want to match. Valid port numbers are from* **0** *to* **255**.
You can also enter well-known TCP or UDP port names as an alternative to the corresponding port number; for example:

- ***TCP***: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- ***UDP***: bootpc, bootps, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To display a list of valid TCP/UDP source ports, type **?** *after you enter an operator.*

[*operator* < *tcp-dest-port* [established] [*tcp-flag* [*tcp-flag* ...]] | *udp-dest-port* >]

To specify a TCP or UDP destination port number as a match criteria, enter a comparison operator with a TDP/UDP port number or well-known port name immediately after the destination-address value in the command.

Note*: The optional* **established** *and* < *tcp-flag* > *values apply only to TCP destination-port criteria.*

TCP/UDP Well-Known Destination-Port Names and Numbers — *The same operators, port numbers and well-known names are supported for TCP/UDP destination-port match criteria as for TCP/UDP source-port criteria. To display a list of valid TCP/UDP destination ports, type* **?** *after you enter an operator.*

[established]

(Optional) Applies only to TCP destination-port match criteria and matches only on the TCP Acknowledge (ACK) or Reset (RST) flags.
The **established** *keyword ignores the synchronizing packet associated with the establishment of a TCP connection in one direction on a port or VLAN, and matches all other IP traffic in the opposite direction.*

For example, a Telnet connection requires TCP traffic to move both ways between a host and the target device. If you configure a match statement for inbound Telnet traffic, policy actions are normally applied to Telnet traffic in both directions because responses to outbound requests are also matched. However, if you enter the **established** *option, inbound Telnet traffic arriving in response to outbound Telnet requests is matched, but inbound Telnet traffic trying to establish a connection is not matched.*

[*tcp-flag* [*tcp-flag ...*]]

(Optional) Applies only to TCP bit settings in packets destined to a TCP destination port configured as match criteria (with the ***operator*** *<* ***tcp-dest-port*** *>parameter) and can be one or more of the following values:*

- **ack** — *"Acknowledge" matches TCP packets with the ACK flag.*
- **fin** — *"Finish" matches TCP packets with the FIN flag.*
- **rst** — *"Reset" matches TCP packets with the RST bit set.*
- **syn** — *"Synchronized" matches TCP packets with the SYN flag.*

Using CIDR Notation for IPv4/IPv6 Addresses

You can use CIDR (Classless Inter-Domain Routing) notation to enter an IPv4 mask-length or an IPv6 prefix-length with a source and destination address that are used as match criteria in a match/ignore statement. The switch interprets the IP address with CIDR notation to compute the range of corresponding IP source or destination addresses in packet headers that are considered to be a match for the traffic class.

When the switch uses a match/ignore statement to compare an IP address and corresponding mask/prefix length to the IP source/destination address carried in a packet, the IPv4 mask-bit settings and IPv6 prefix-bit settings select packets in different ways.

- An IPv4 mask length creates a mask in which:
  - A mask-bit setting set to **0** ("off") requires the corresponding bit in a packet's IPv4 source/destination address to be the same binary value as the mask-bit in the matching IPv4 source/destination address.
  - A mask-bit setting set to **1** ("on") is used as a wildcard and allows the corresponding bit in a packet's IPv4 source/destination address to be either binary value (0 or 1).

Table 9-1. How CIDR Notation is Used with IPv4 SA/DA Match Criteria

IPv4 Source/Destination Address Used with CIDR Notation in a Match/Ignore Statement	Resulting Mask	Range of IPv4 Addresses Selected by the Match Criteria
10.38.240.125/15	0.1.255.255	The leftmost 15 bits must match; the remaining bits are wildcards.
10.38.240.125/20	0.0.15.255	The leftmost 20 bits must match; the remaining bits are wildcards.
10.38.240.125/21	0.0.7.255	The leftmost 21 bits must match; the remaining bits are wildcards.
10.38.240.125/24	0.0.0.255	The leftmost 24 bits must match; the remaining bits are wildcards.
18.38.240.125/32	0.0.0.0	All bits must match.

- An IPv6 prefix-length creates a mask in which:
  - A mask-bit setting set to **1** ("on") requires the corresponding bit in a packet's IPv6 source/destination address to be the same binary value as the mask-bit in the matching IPv6 source/destination address.
  - A mask-bit setting set to **0** ("off") is used as a wildcard and allows the corresponding bit in a packet's IPv6 source/destination address to be either binary value (0 or 1).

Table 9-2. How CIDR Notation is Used with IPv6 SA/DA Match Criteria

IPv6 Source/Destination Address Used with CIDR Notation in a Match/Ignore Statement	Resulting Mask	Range of IPv6 Addresses Selected by the Match Criteria
2001:db8:0:7::5/64	FFFF:FFFF:FFFF:FFFF::	The leftmost 64 bits must match; the remaining bits are wildcards.
2001:db8:0:7::5/72	FFFF:FFFF:FFFF:FFFF:FF00::	The leftmost 72 bits must match; the remaining bits are wildcards.
2001:db8::244:17ff:feb6:d37d /126	FFFF:FFFF:FFFF:FFFF:FFFF: FFFF:FFFF:FFFC	The first 126 bits mst match; the "C" value in the mask allows four possible combinations (D37C, D37D, D37E, and D37F) in the last block of a matching IPv6 address.
2001:db8:0:7:af:e2:c1:5/128	FFFF:FFFF:FFFF:FFFF:FFFF: FFFF:FFFF:FFFF	All bits must match.

Note

Although IPv4 and IPv6 masks are applied in opposite directions:

- An IPv4 mask-length is applied from right to left, starting from the rightmost bits.
- An IPv6 prefix-length is applied from left to right, starting from the leftmost bits.

The behavior of IPv4 and IPv6 masks as match criteria and wildcards is the same.

Example of How IPv4 Mask Bit Settings Define a Match. For this example, the following configuration exists:

- A match statement in a class configuration uses an IPv4 source-address/mask-length of 10.38.31.125/21. The mask-length of 21 results in an IPv4 mask of 0.0.7.255. In the second octet of the mask, 7 means that the rightmost three bits are "on", or "1" (see the "Mask for SA" row in Table 9-3).
- The second octet of the corresponding source address is 31, which means that the rightmost five bits are "on", or "1" (see the SA in Match Statement row in Table 9-3).

In this example, a match occurs when the second octet of the SA in a packet being classified has a value in the range of 24 (binary 00011000) to 31 (binary 00001111), as shown in the last row in Table 9-3.

Table 9-3. Example of How the IPv4 Mask Defines a Match

Location of Octet	Bit Position in the Octet							
	128	64	32	16	8	4	2	1
SA in match statement	0	0	0	1	1	1	1	1
Mask for SA	0	0	0	0	0	1	1	1
Bits in the corresponding octet of a packet's SA that must exactly match	0	0	0	1	1	0/1	0/1	0/1

The shaded area indicates the bits in the packet that must exactly match the bits in the source IPv4 address in the match/ignore statement.

- If a mask bit is "1"(wildcard value), the corresponding bits in a source/destination address in an IPv4 packet header can be any value.
- If a mask bit is "0", the corresponding bits in a source/destination address must be the same value as in the IPv4 address in the match/ignore statement.

Note: This example covers only one octet in an IPv4 address used as a match criterion. The mask in a match/ignore statement may apply a packet filter to all four octets of a source/destination address in IPv4 packet headers.

Example of How IPv6 Mask Bit Settings Define a Match. Figure 9-4 shows an example in which an IPv6 prefix-length of 126 is used to select four IPv6 addresses in a match statement. The specified source IPv6 address is: **2001:DB8:0000:0000:244:17FF:FEB6:D37D**. The IPv6 prefix-length (/126) results in the IPv6 mask: **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC**.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 mask for /126 prefix	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC	The "F" value in the first 126 bits of the mask specifies that only the exact value of each corresponding bit in an IPv6 address is allowed. However, the binary equivalent (1100) of the "C" value in the mask allows four possible combinations (D37C, D37D, D37E, and D37F) in the last block of a matching IPv6 address.
IPv6 address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

Figure 9-4. Example: Mask for Matching Four IPv6 Devices

Figure 9-5 shows the "on" and "off" settings in the last block of the resulting IPv6 mask that determine the matching IPv6 addresses. In this mask, all bits except the last two are set to 1 ("on") and must be the same in an IPv6 address. The binary equivalent of hexadecimal **C** is 1100, which allows the last two bits to differ.

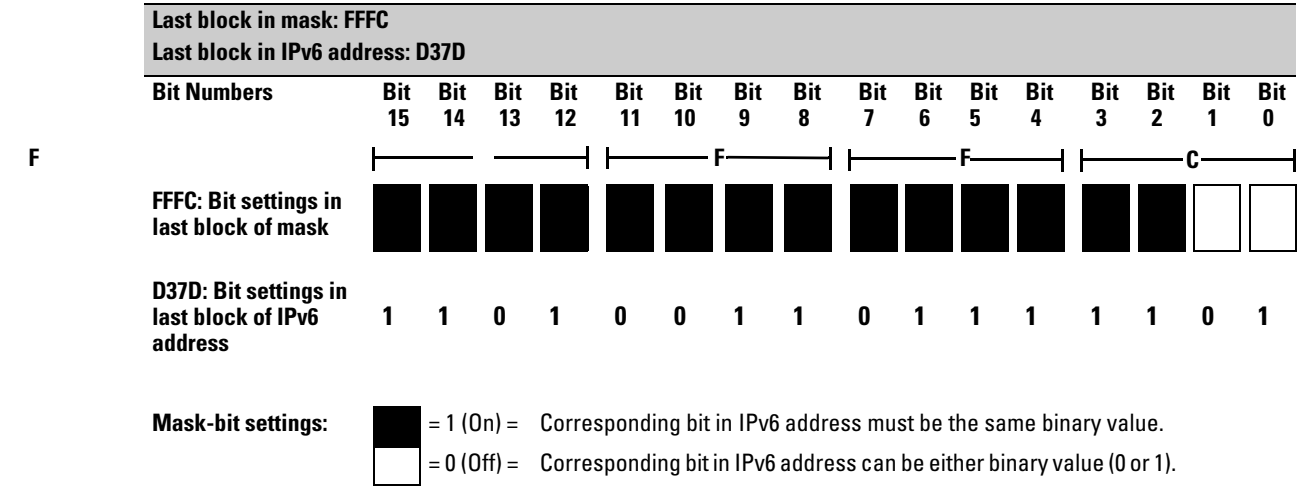

Last block in mask: FFFC
Last block in IPv6 address: D37D
Bit Numbers
Bit 15 Bit 14 Bit 13 Bit 12 Bit 11 Bit 10 Bit 9 Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0
F
F
F
C
FFFC: Bit settings in last block of mask
D37D: Bit settings in last block of IPv6 address
1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1
Mask-bit settings:
= 1 (On) = Corresponding bit in IPv6 address must be the same binary value.
= 0 (Off) = Corresponding bit in IPv6 address can be either binary value (0 or 1).

Figure 9-5. Example: How a Mask Determines Four Authorized IPv6 Manager Addresses

Figure 9-6 shows how the binary equivalent (1100) of the "C" value in the last block of the resulting IPv6 mask supports four possible combinations (D37C, D37D, D37E, and D37F) in the last block of a matching IPv6 address. Therefore, the IPv6 mask that results from a /126 prefix-length matches inbound traffic from four IPv6-based devices.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block
IPv6 mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC
IPv6 address entered with a "match" command	2001	DB8	0000	0000	244	17FF	FEB6	D37D
Other matching IPv6 addresses	2001	DB8	0000	0000	244	17FF	FEB6	D37C
	2001	DB8	0000	0000	244	17FF	FEB6	D37E
	2001	DB8	0000	0000	244	17FF	FEB6	D37F

Figure 9-6. Example: How Hexadecimal C in an IPv6 Mask Matches Four IPv6 Addresses

CIDR Notation. For more detailed information on how to use CIDR notation to specify masks in match criteria, refer to the "How an ACE Uses a Mask To Screen Packets for Matches" section in the *Access Control Lists (ACLs)* chapter in the *Access Security Guide*.

Resequencing Match/Ignore Statements

In the class configuration context (see "Creating a Traffic Class" on page 9-4), you can use the **resequence** command to reconfigure the number at which the first match/ignore statement in the class starts, and reset the interval used to number other match/ignore statements.

Resequencing match/ignore statements is useful when you want to insert a new match/ignore statement between two numbered entries (see Figure 9-7).

Context: Class configuration

Syntax: resequence < *seq-number* > < *interval* >

Resets the sequence numbers for all match/ignore statements in the class.

< *seq-number* > : *Specifies the sequence number of the first match/ignore statement in the class. Default: 10.*

< *interval* > : *Specifies the interval between sequence numbers of match/ignore statements in the class to allow additional match/ignore statements to be inserted. Default: 10.*

To view the current sequence numbering in a class, enter the **show class <ipv4 | ipv6 > < *classname* >** command.

The following example shows how to resequence a class configuration so that you can insert new match/ignore statements between sequentially numbered statements. In this example, the resequenced class contains two additional match/ignore statements and renumbers the criteria with an interval of 10.

```
ProCurve(config)# show class ipv4 My-devices

Class “My-devices"
   1 match 10.10.10.25 0.0.0.0
   2 ignore 10.10.10.1 0.0.0.255
   3 ignore 10.20.10.1 0.0.0.255
   4 match 0.0.0.0 255.255.255.255
   exit
. . .
ProCurve(config)# class ipv4 My-devices
ProCurve(config-class)# resequence My-devices 10 10
ProCurve(config-class)# 15 match 10.10.10.2 0.0.0.255
ProCurve(config-class)# 25 match 10.20.10.1 0.0.0.255
ProCurve(config-class)# exit
ProCurve(config)# show class My-devices

Class “My-devices"
   10 match 10.10.10.25 0.0.0.0
   15 match 10.10.10.2 0.0.0.255
   20 ignore 10.10.10.1 0.0.0.255
   25 ignore 10.20.10.1 0.0.0.255
   30 ignore 10.20.10.2 0.0.0.255
   40 match 0.0.0.0 255.255.255.255
   exit
```

The interval between match/ ignore statements is 1.

The interval between match/ ignore statements is 10 and two new match/ignore statements have been added.

Figure 9-7. Example of Resequencing a Class Configuration

Creating a Service Policy

In the *classifier-based* configuration model, the service policy you create for one or more traffic classes is always relative to a software feature, such as QoS or port mirroring. The software feature must support class and policy configuration. Each feature supports different actions for managing selected packets.

For example, QoS policies support QoS-specific actions, such as rate-limiting, 802.1p-priority, IP-precedence, and DSCP-codepoint assignment. Port mirroring policies support mirror-destination assignment for matching packets.

1. To create a service policy that performs feature-specific actions on selected packets, enter the **policy < *feature-name* >** command from the global configuration context.

Context: Global configuration

Syntax: [no] policy <*feature-name* > <*policy-name* >

Defines the name of a service policy and enters the policy configuration context, where:
<***feature-name*** > *is a keyword that identifies a ProCurve software feature that supports classifier-based configuration (for example,* **qos** *or* **mirror***).*
<***policy-name*** > *is a text string (64 characters maximum).*

A traffic policy consists of one or more actions that are configured for each class of traffic. The configured actions are executed on packets that match a **match** statement in a class. No policy action is performed on packets that match an **ignore** statement.You can configure multiple classes in a policy.

2. To configure the actions that you want to execute on packets that match the **match** criteria in a specified class, enter one or more **class action** commands from the policy configuration context.

Context: Policy configuration

Syntax: [no] [*seq-number*] class < ipv4 | ipv6 > <*classname* >
action <*action-name* > [action <*action-name* > ...]

Defines the action(s) to be applied on a pre-configured IPv4 or IPv6 traffic class when a packet matches the **match** *criteria in the class.*

You can enter multiple class-action statements for the same class. Note that the actions supported for a class command differ according to the feature-specific policy (for example, QoS or mirroring) configured with the **policy** *command in Step 1.*

[no] [*seq-number*] class < ipv4 | ipv6 > <*classname* >

- [***seq-number***] — *The (optional)* ***seq-number*** *parameter sequentially orders the class-action statements in a policy configuration. Actions are executed on matching packets in numerical order. Default: Class-action statements are numbered in increments of 10, starting at 10.*

- **class** < **ipv4** | **ipv6** > <*classname*> — *Defines the preconfigured class on which the actions in a class-action statement are executed, and specifies whether the class consists of IPv4 or IPv6 traffic. The class name is a text string (64 characters maximum).*

***Note**: You can configure multiple class-action statements to include different classes in a policy. The execution of actions is performed in the order in which the class-actions are numerically listed.*

action <*action-name*> [action <*action-name*> ...]

The **action** *keyword configures the action specified by the* ***action-name*** *parameter. The action is executed on any packet that matches the* **match** *criteria in the class. The action is not executed on packets that match* **ignore** *criteria. You can configure more than one action for a class.*

The complete **no** *form of the* **class action** *command or the* **no <*seq-number*>** *command removes an action from the policy configuration.*

For information on the exact actions supported by each classifier-based software feature, refer to the appropriate chapter in the ProCurve documentation set, as described in "" on page 9-33.

Be sure to enter a class and its associated actions in the *precise order* in which you want packets to be checked and handled by **class action** commands.

3. (Optional) To configure a default class, enter the **default-class** command and specify one or more actions to be executed on packets that are not matched and not ignored.

Context: Policy configuration

Syntax: [no] default-class action <*action-name*> [action <*action-name*> ...]

Configures a default class to be used to execute one or more actions on packets that are not matched nor ignored in any of the class configurations in a policy.

The **default-class action** *command supports only the feature-specific commands supported in the* **class action** *command.*

The default class manages packets that do not match the **match** or **ignore** criteria in all classes in a policy, and otherwise would have no actions performed on them.

The default class differs from other classes because it contains no match/ignore statements and uses implicit **match ipv4 any any** and **match ipv6 any any** statements to manage all unmatched packets. If you do not configure a default class, unmatched and unignored packets are transmitted without an action performed on them.

4. Enter the **exit** command to exit the policy configuration context.

   To display a policy configuration, enter the **show policy < *feature-name* > < *policy-name* >** command (see Figure 9-9), where **< *feature-name* >** is a software feature (such as QoS or port mirroring) that supports classifier-based configuration.

   To edit a policy configuration, re-enter the policy context (**policy** command) and modify class-action statements as described in "Modifying Classes in a Policy" on page 9-28.

   To resequence the order in which class-action statements are listed, enter the **resequence** command (see "Resequencing Classes in a Policy" on page 9-29).

Example. In the following QoS policy configuration, matching HTTP packets are rate-limited to 10000 kbps and assigned an 802.1p (CoS) priority of 3 in their Layer 2 VLAN headers. All unmatched packets are managed by the default class, which assigns a slightly higher 802.1p priority (4) and a new DSCP codepoint (5)

```
ProCurve(config)# class ipv4 http
ProCurve(class-config)# match tcp any any eq 80
ProCurve(class-config)# match tcp any any eq 8080
ProCurve(class-config)# exit
ProCurve(config)# policy qos RateLimitPrioritizeSuspectTraffic
ProCurve(policy-config)# class ipv4 http action rate-limit kbps 10000
ProCurve(policy-config)# class ipv4 http action priority 3
ProCurve(policy-config)# default-class action priority 4 dscp 5
ProCurve(policy-config)# exit
```

Figure 9-8. Example of a Policy Configuration

As shown in Figure 9-8, a policy configuration requires a feature-specific **policy** command to identify the software feature used to manage one or more traffic classes:

- To configure a QoS policy, use the **policy qos** command as described in the "Quality of Service" chapter in the *Advanced Traffic Management Guide*.

- To configure a mirroring policy, use the **policy mirror** command as described in the *Monitoring and Analyzing Switch Operation* appendix in the *Management and Configuration Guide*.

Modifying Classes in a Policy

You can modify the classes and class-action statements in a policy configuration without removing them from the policy:

- To modify the match/ignore statements in a class, enter the class-configuration context with the **class <ipv4 | ipv6> <*classname*>** command, and make the necessary changes by removing, resequencing, or replacing existing statements. (To display a class configuration, enter the **show class < ipv4 | ipv6 > < *classname* >** command as shown in Figure 9-7.)

  When you exit class configuration context, the changes are automatically saved and applied to existing policy configurations on the switch that use the class *if the policies have not been applied to an interface*. If a policy has already been applied to an interface, the editing changes are not accepted and an error message is displayed.
- To modify the class-action statements in a policy, enter the policy-configuration context with the **policy <*feature-name*> <*policy-name*>** command. (To display a policy configuration, enter the **show policy < *feature-name* > < *classname* >** command as shown in Figure 9-9.) Then do one of the following:
  - You can enter a new class-action statement. If you do not enter a sequence number, the new class-action statement is inserted at the end of the policy configuration.
  - To remove a class-action statement from a policy configuration, enter the **no** <***sequence-number***> command or the complete form of the **no class ... action** command.
  - To resequence the order in which class-action statements are listed, enter the **resequence** command (see "Resequencing Classes in a Policy" on page 9-29).
  - To replace an existing class-action statement, enter the **no** <***sequence-number***> command to delete the entry, and re-enter a complete **class < ipv4 | ipv6 > < *classname* > action < *action-name* >** or **default-class action < *action-name* >** command.

  When you exit the policy-configuration context, the changes are automatically saved and applied to the policy configuration *if the policy has not been applied to an interface*. If the policy has already been applied to an interface, the editing changes are not accepted and an error message is displayed.

Resequencing Classes in a Policy

In policy configuration mode, you can use the **resequence** command to reconfigure the number at which the first class-action statement starts, and reset the interval used to number other class-actions.

Resequencing classes is useful when you want to insert a new class (with its associated actions) between two numbered entries.

Context: Policy configuration

Syntax: resequence < *seq-number* > < *interval* >

Resets the sequence numbers for all classes in the policy.
< *seq-number* > : *Specifies the sequence number of the first class in the policy. Default: 10.*
< *interval* > : *Specifies the interval between sequence numbers of classes in the policy to allow additional match/ignore statements to be inserted. Default: 10.*

Note

When you resequence classes in a policy, the default class always remains as the last class.

To view the current class numbering in a policy, enter the **show policy < *feature-name* > < *policy-name* >** command.

The following example shows how to resequence a policy configuration after displaying its contents. The resequenced policy allows you to add a new class-action statement between entries 100 and 200.

```
ProCurve(config)# show policy My-devices
Policy “My-devices"
   10 Class “My-devices" priority 7
   11 Class “http” rate-limit 1000
...
ProCurve(config)# policy My-devices
ProCurve(policy-config)# resequence My-devices 100 100
ProCurve(policy-config)# 150 class ipv4 voice priority 3
ProCurve(policy-config)# exit
ProCurve(config)# show policy My-devices

Policy “My-devices"
   100 Class “My-devices" priority 7
   150 Class “voice” priority 3
   200 Class “http” rate-limit 1000
```

The interval between class-action statements is 1.

The interval between class-action statements is 100, and a new statement has been added.

Figure 9-9. Example of Resequencing a Policy Configuration

Applying a Service Policy to an Interface

To apply the feature-specific service policies you create to an inbound port or VLAN interface, use the **interface service-policy in** or **vlan service-policy in** command.

The following service-policy restrictions apply to all software features:

- A service policy is supported only on inbound traffic.
- Only one feature-specific policy (for example, QoS or mirroring) is supported on a port or VLAN interface.
- If you apply a policy to a port or VLAN interface on which a policy of the same type (for example, QoS) is already configured, an error message is displayed. The new policy does not overwrite the existing one.

  Before you can apply a new policy, you must first remove the existing policy with the **no interface service-policy in** or **no vlan service-policy in** command.

Because only one policy of each type is supported on a port or VLAN interface, ensure that the policy you want to apply contains all the required classes and actions for your configuration.

Note

If ICMP rate-limiting is already configured on a port, a service policy cannot be applied to the port until you disable the ICMP rate-limiting configuration.

If you want to apply a service policy to the port, you can maintain ICMP rate-limiting by configuring a QoS policy in which you add the necessary **match** statements for ICMP packets to a class configuration and configure a **rate-limit** action for the class in the policy configuration.

For information on globally-configured ICMP, refer to the "Configuring ICMP" section in the "Configuring IP Parameters for Routing Switches" chapter in the *Multicast and Routing Guide*.

To apply a service policy on a port or VLAN interface, enter one of the following commands from the global configuration context.

Context: Global configuration

Syntax: interface <*port-list*> service-policy <*policy-name*> in

Configures the specified ports with a policy that is applied to inbound traffic on each interface.

Separate individual port numbers in a series with a comma; for example, **a1, b4, d3**. *Enter a range of ports by using a dash; for example,* **a1-a5**.

The policy name you enter must be the same as the policy name you configured with the **policy** *command (see "Creating a Service Policy" on page 9-24).*

Context: Global configuration

Syntax: vlan <*vlan-id*> service-policy <*policy-name*> in

Configures a policy on the specified VLAN that is applied to inbound traffic on the VLAN interface.

Valid VLAN ID numbers range from **1** *to* **4094**.

The policy name you enter must be the same as the policy name you configured with the **policy** *command (see "Creating a Service Policy" on page 9-24).*

The following example shows how to apply a QoS policy to a port range and a VLAN interface:

```
ProCurve# interface a4-a5 service-policy RateLimitPrioritizeSuspectTraffic in
ProCurve# vlan 10 service-policy RateLimitPrioritizeSuspectTraffic in
```

Figure 9-10. Example of How to Configure an Interface with a Service Policy

Checking Resource Usage. After you apply a service policy to an interface, use the **show policy resources** command to verify the amount of additional resources used and the amount of resources that are still available on the switch. Classifier-based service policies (such as QoS or mirroring) share the same hardware resources with other software features, such as ACLs, virus throttling, management VLAN, globally configured QoS policies, MAC-based mirroring policies, and so on.

Use the displayed information to decide if you need to re-prioritize current resource usage by reconfiguring or disabling software features to free the resources reserved for less important features. For a detailed explanation of

the information displayed with the **show policy resources** command, refer to the "Monitoring Resources" appendix in the *Management and Configuration Guide*.

In Figure 9-11, the **show policy resources** command output displays the number of hardware resources (rules, meters, and application port ranges) used by classifier-based QoS and mirroring policies that are currently applied to interfaces on the switch, as well as other software features.

```
ProCurve# show policy resources

 Resource usage in Policy Enforcement Engine
        |    Rules     |  Rules Used
  Ports |  Available   |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+--------------+-------+-------+-------+-------+--------+-------|
  1-24  |         3014 |    15 |    11 |     0 |     1 |      0 |     3 |
  25-48 |         3005 |    15 |    10 |    10 |     1 |      0 |     3 |
  A     |         3017 |    15 |     8 |     0 |     1 |      0 |     3 |

        |    Meters    |  Meters Used
  Ports |  Available   |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+--------------+-------+-------+-------+-------+--------+-------|
  1-24  |          250 |       |     5 |     0 |       |        |     0 |
  25-48 |          251 |       |     4 |     0 |       |        |     0 |
  A     |          253 |       |     2 |     0 |       |        |     0 |

        | Application |
        | Port Ranges |  Application Port Ranges Used
  Ports |  Available   |  ACL  |  QoS  |  IDM  |   VT  | Mirror | Other |
  ------+--------------+-------+-------+-------+-------+--------+-------|
  1-24  |         3014 |     2 |     0 |     0 |       |      0 |     0 |
  25-48 |         3005 |     2 |     0 |     0 |       |      0 |     0 |
  A     |         3017 |     2 |     0 |     0 |       |      0 |     0 |

 0 of 8 Policy Engine management resources used.
 Key:
 ACL = Access Control Lists
 QoS = Device & Application Port Priority, QoS Policies, ICMP rate limits
 IDM = Identity Driven Management
 VT  = Virus Throttling blocks
 Mirror = Mirror Policies, Remote Intelligent Mirror endpoints
 Other = Management VLAN, DHCP Snooping, ARP Protection, Jumbo IP-MTU.

 Resource usage includes resources actually in use, or reserved for future
 use by the listed feature.  Internal dedicated-purpose resources, such as
 port bandwidth limits or VLAN QoS priority, are not included.
```

Includes hardware resources used by classifier-based QoS and mirroring policies that are currently applied to interfaces on the switch.

Figure 9-11. Displaying Policy Resources

Where to Go From Here

Classifier-based service policies are designed to work with your existing globally-configured software settings. While existing software features allow you to globally manage *all* network traffic on a switch or port, classifier-based service policies allow you to zoom in on subsets of network traffic to further manage it on a per-port or per-VLAN basis.

You can use the match criteria described in this chapter across software features to configure classes of traffic for use in feature-specific service policies.

After you decide on the IPv4 and IPv6 network traffic you want to manage, refer to the following chapters for more information about how to configure and use classifier-based quality-of-service and mirroring policies:

- *Quality of Service (QoS)* chapter in the *Advanced Configuration Guide*
- "Traffic Mirroring" section in the *Monitoring and Analyzing Switch Operation* appendix in the *Management and Configuration Guide*

Index

Numerics

802.1ad QinQ … 8-4, 8-6
802.1p priority
definition … 6-7
determining outbound port queue … 6-76
in DSCP policy … 6-8
mapped to DSCP codepoint … 6-79, 6-88
packet marking
classifier-based QoS … 6-18, 6-71, 6-78
global QoS … 6-14, 6-25, 6-34, 6-44, 6-54, 6-56, 6-62
802.1Q VLAN tagging … 3-3
802.1w as a region … 4-16
802.1X
mesh, not supported … 5-5
802.1X access control
no mesh … 5-5

A

advertisement
GVRP
definition … 3-3
alias
with show VLAN commands … 2-37

B

bandwidth
effect of QoS … 6-1
bandwidth loss, spanning tree … 4-12
blocked link from STP operation … 4-12
blocked port
from STP operation … 4-10
Bootp
gateway ignored … 2-52
BPDU … 3-3
BPDU port protection
See spanning-tree, 802.1s.
bridge protocol data unit … 3-3
broadcast
domain … 2-5
broadcast storm … 4-3, 5-4
broadcast traffic … 5-16

C

Class of Service
defined … 6-7
Classifier
benefits … 6-6, 6-70, 9-2
class configuration
commands … 9-5
defined … 9-4
displaying … 9-12, 9-23
editing … 9-12
match criteria … 6-11, 6-13, 6-71, 6-72
QoS … 6-7
resequencing … 9-12, 9-23
default-class configuration … 6-75, 9-5, 9-26
defined … 6-7, 9-2
DSCP codepoint, match criteria … 9-9
ICMP match criteria … 9-13
IGMP match criteria … 9-16
IP precedence bits, match criteria … 9-10
IPv4 subnet mask, match criteria … 9-19
IPv6 match criteria … 9-7
IPv6 prefix, match criteria … 9-19
mirroring configuration … 9-3, 9-33
override of global QoS configuration … 6-18, 6-70, 6-81
packet marking
802.1p priority … 6-18, 6-71, 6-78
DSCP codepoint … 6-18, 6-71, 6-79
precedence bits … 6-18, 6-71, 6-78
rate-limit … 6-18, 6-77
rate-limiting … 6-71
policy configuration
applying to an interface … 6-75, 9-30
commands … 9-31
creating … 9-25
defined … 9-24
displaying … 9-27, 9-29
editing … 9-27, 9-28
example … 9-27
marking packets … 6-8, 6-11, 6-71, 6-73, 9-24
QoS … 6-8, 9-27
resequencing … 9-27, 9-29
restrictions … 9-30

QoS configuration … 6-6, 6-70, 6-71, 9-3, 9-27, 9-33
examples … 6-86
resources used … 6-85
viewing … 6-81
resource usage, displaying … 6-83, 9-31
restrictions
ICMP rate-limiting … 9-30
QoS configuration … 6-75, 6-85
QoS rate-limiting … 6-77
TCP/UDP match criteria … 9-17
three-step configuration model … 6-7, 9-2
Type of Service DTR bits, match criteria … 9-10
VLAN match criteria … 9-11
codepoint
See DSCP codepoint.
command syntax conventions … 1-2
configuration … 4-10
factory default … 2-24, 2-30, 4-9
spanning tree protocol … 4-10
console
switch meshing, configuring … 5-9
CoS
See Class of Service.

D

dedicated management VLAN … 2-51
default class, classifier … 6-75, 9-5, 9-26
default settings
GVRP unknown VLAN, *learn* … 3-8
GVRP, *disabled* … 3-3
ip-recv-mac-address interval, *60 seconds* … 2-68
loop protection, *disabled* … 4-75
loopback IP address, *127.0.0.1* … 2-8
MAC address, restoring … 2-68
management VLAN, *disabled* … 2-55
MSTP … 4-3
default settings recommended … 4-6, 4-9
MST instance, *0* … 4-70
See also spanning tree. … 4-3
QinQ, *disabled* … 8-7
spanning tree … 4-3
config-name … 4-22
config-revision, *0* … 4-22
forward-delay, *15* … 4-23
hello-time, *2* … 4-24
max-hops, *20* … 4-19, 4-24
maximum age, *20* … 4-24
path cost values … 4-17
per port parameters … 4-26–4-30
pvst-filter, *disabled* … 4-36
pvst-protection, *disabled* … 4-35
spanning tree operation, *off* … 4-9
stacking … 7-4, 7-10
auto join, *enabled* … 7-15
stack state, *candidate* … 7-15
transmission interval, *60 seconds* … 7-15
stacking, *enabled* … 7-29
static VLANs, *8* … 2-5
VLAN name … 2-31, 2-41
vlan qos priority, *0* … 2-61
VLAN support, *256 VLANs* … 2-19
default VLAN … 2-10, 2-15, 2-30, 2-51
Delay Throughput Reliability bits
in IPv4/IPv6 headers … 9-11
match criteria, classifier
DHCP
gateway ignored … 2-52
Differentiated Services (DiffServ)
See DSCP codepoint.
documentation
feature matrix … -xvi
latest versions … -xv
release notes … -xv
domains, connecting … 5-24
downstream device (QoS)
definition … 6-7
effect of 802.1p priority settings … 6-15
DSCP codepoint
associated with 802.1p priority … 6-8
defined … 6-7, 6-42
determining outbound port queue … 6-53
in IPv4/IPv6 headers … 9-11
in ToS byte … 6-9, 6-43
in Traffic Class byte … 6-9, 6-43
mapped to 802.1p priority … 6-88
match criteria, classifier … 6-72, 9-9
match criteria, global QoS … 6-41, 6-45
packet marking
classifier-based QoS … 6-18, 6-71, 6-79
global QoS … 6-16, 6-27, 6-49, 6-58, 6-64
policy, defined … 6-8
See also 802.1p priority.
DSCP Policy table … 6-79, 6-89
DTR

See Delay Throughput Reliability bits.

E

event log
Event Log Message Reference Guide … -xv

F

forbid option
See GVRP.
forwarding database
See VLAN.

G

GARP
See GVRP.
gateway
manual config priority … 2-52
GVRP … 4-8
ACLs, restriction … 3-19
advertisement … 3-19
advertisement, defined … 3-3
advertisement, responses to … 3-6
advertisements, generating … 3-11
auto option … 3-10
benefit … 3-3
block … 3-8
CLI, configuring … 3-14
configurable port options … 3-6
configuring learn, block, disable … 3-8
convert dynamic to static … 3-7
converting to static VLAN … 3-4
disable … 3-8
dynamic VLAN and reboots … 3-19
dynamic VLANs always tagged … 3-4
forbid option … 3-10
GARP … 3-3
general operation … 3-4
IP addressing … 3-7
jumbo frames … 3-19
learn … 3-8
learn, block, disable … 3-10
menu, configuring … 3-13
meshed ports … 5-21
meshing requirement … 5-6
non-GVRP aware … 3-18
non-GVRP device … 3-18
operating notes … 3-18
port control options … 3-11
port-leave from dynamic … 3-11
provider-gvrp (QinQ) … 8-15
reboot, switch … 3-12
recommended tagging … 3-11
standard … 3-3
tagged, dynamic VLAN … 3-4
unknown VLAN … 3-11
unknown VLAN, options … 3-7
VLAN behavior … 2-14
VLAN, dynamic adds … 2-28
VLAN, maximum … 3-18
with QinQ … 8-29
with QoS … 6-56

H

heartbeat packets in VLAN MAC configuration … 2-67
Help
for CLI … 1-7
for menu interface … 1-6
for web browser interface … 1-7

I

IANA, protocol numbers … 9-7, 9-13
ICMP
match criteria, classifier … 9-13
IEEE 802.1ad specification … 8-4, 8-6
IGMP
in switch mesh domain … 5-20
match criteria, classifier … 9-16
mesh requirement … 5-6
inbound port (QoS)
definition … 6-8
IP
gateway … 2-52
IP address
match criteria, classifier … 6-72, 9-7
match criteria, global QoS … 6-33
quick start … 1-7
IP precedence bits
defined … 6-8, 6-42
determining outbound port queue … 6-53
in IPv4/IPv6 headers … 9-11

in ToS byte … 6-9, 6-43
in Traffic Class byte … 6-43
match criteria
classifier … 6-72, 9-10
global QoS … 6-41
IPv4
DSCP codepoint, defined … 6-7
mask used in match criteria, classifier … 9-19, 9-21
IPv6
class configuration, classifier … 9-5
DSCP codepoint, defined … 6-7
IP precedence bits in Traffic Class byte … 6-9
management VLAN is for v6 traffic … 2-59
match criteria
classifier … 6-6, 6-11, 6-13, 6-70, 6-72
classifier-based QoS … 6-34
global QoS … 6-22, 6-33
overview … 9-5
prefix mask … 9-9, 9-19, 9-22
Traffic Class compared to IPv4 ToS field … 9-11

J

jumbo frames
GVRP … 3-19
QinQ … 8-11
switch mesh … 5-21

L

LACP
effects on QinQ … 8-23
mesh, effect … 5-5
latency
reducing with switch meshing … 5-17
Layer-3 protocol
match criteria, classifier … 6-72
match criteria, global QoS … 6-54
legacy VLAN … 2-13
link failures
quick response to … 5-2
links, redundant, in mesh … 5-24
loop protection
configuring … 4-74
send-disable … 4-74
show … 4-76
transmit-interval … 4-75
trap … 4-75
used for unmanaged devices … 4-3
loop, network … 4-10

M

MAC address
duplicate … 2-20
per switch … 2-20
per VLAN … 2-20
same for all VLANs … 2-63
single forwarding database … 2-20
management VLAN, secure
See secure management VLAN
marking packets … 6-8
802.1p priority … 6-14, 6-18
classifier-based QoS … 6-18
DSCP codepoint … 6-16, 6-18
global QoS … 6-14
precedence bits … 6-18
rate-limit … 6-18
match criteria
classifier-based QoS … 6-13, 6-71, 6-72
classifier-based, defined … 9-4
DSCP codepoint … 6-41, 9-9
global QoS … 6-12
ICMP … 9-13
IGMP … 9-16
IP address … 6-33, 9-7
IP precedence bits … 6-41, 9-10
Layer-3 protocol … 6-54
source port … 6-62
TCP/UDP … 6-24, 9-17
ToS bits … 9-10
Type of Service … 6-41
VLAN ID … 6-56, 9-11
maximum VLANs, GVRP … 3-18
mesh
802.1X not supported … 5-5
benefits … 5-2
blocked ports … 5-8
broadcast storm … 5-4
broadcast traffic … 5-16
broadcast tree … 5-16
configuring from the console … 5-9
connecting domains … 5-24
connecting multiple domains … 5-6
domain … 5-3

domain, defined … 5-4
dynamic vlan … 5-21
edge switch … 5-4, 5-16
filtering … 5-20
GVRP … 5-21
GVRP requirement … 5-6
hop count … 5-5
hub not allowed … 5-5, 5-7
IGMP requirement … 5-6
increase STP cost … 5-19
jumbo frames … 5-21
LACP dynamic trunk, effect … 5-5
link blocked … 5-19
link to non-mesh switch … 5-18
links, multiple … 5-24
load-balancing technology … 5-2
management VLAN … 2-59
multicast traffic … 5-16
multiple mesh domains … 5-19
multiple VLANs … 5-17
no Type selection … 5-24
operating details … 5-15
operating notes … 5-15
operating rules … 5-5
port limit per-switch … 5-5
port trunk … 5-24
port types … 5-2
QinQ … 8-12, 8-30
redundant links … 5-4, 5-19, 5-24
redundant paths … 5-3
removing a port, effect … 5-5
RSTP … 5-6
RSTP caution … 5-19
spanning tree … 4-16
spanning-tree requirement … 5-6
static VLANs … 5-20
status, viewing … 5-12
STP … 5-6
STP caution … 5-19
switch hop count … 5-23
switch limit per-domain … 5-5
trunked links not allowed … 5-5, 5-7
Type setting … 5-10
unicast … 5-17
utilization … 5-15
VLAN … 5-20
VLAN, dynamic … 5-6
VLAN, static … 5-6
when IP routing is not supported … 5-5
with IGMP … 5-20
with network monitor port … 5-24
message
VLAN already exists … 2-44
mirroring
classifier-based configuration … 9-33
MSTI, configuration … 4-39
MSTP
instance mapping … 4-47
meshing … 5-17
preconfigure benefits … 4-46
preconfigure topology … 4-46
preconfigure vlans in instance … 4-47
saving current configuration … 4-51
See spanning-tree, 802.1s.
vlan range option … 4-48
multicast traffic … 5-16
multiple forwarding database … 2-20

N

non-routable VLAN … 2-59

O

operating notes
switch meshing … 5-15
outbound port (QoS)
definition … 6-8
outbound port queue (QoS)
changing the number of queues … 6-96
definition … 6-8
determined by 802.1p priority … 6-5, 6-53
determined by DSCP policy … 6-5, 6-53

P

path costs
802.1D STP versus RSTP and MSTP … 4-17
configuring 802.1D STP pathcost values … 4-23
port
blocked by STP operation … 4-10
blocked in mesh … 5-8
loop … 4-10
manually re-enabling … 4-36
match criteria, global QoS … 6-62
monitoring … 2-63

redundant path … 4-10
port trunk
meshed switch … 5-24
VLAN … 2-63
port-type (QinQ)
customer-edge port … 8-12
customer-network port … 8-6
provider-network port … 8-6
precedence bits (QoS)
definition … 6-8
packet marking
classifier-based QoS … 6-18, 6-71, 6-78
Premium License
overview, list of features … -xvi
QinQ … 8-3
primary VLAN
See VLAN, primary.
priority
802.1p priority, defined … 6-7
associated with DSCP … 6-7
configuring number of queues … 6-95
downstream device, defined … 6-7
inbound port … 6-8
outbound port … 6-8
packet marking
classifier-based QoS … 6-18
global QoS … 6-14
queues per port … 6-95
upstream device … 6-9
priority (QoS)
changing queues per port … 6-95
IP address, source and destination match … 6-34
match criteria for prioritizing packets … 6-12
VID, effect of eliminating … 6-56
ProCurve
switch documentation … -xv
PVST
disabling … 4-35
enabling … 4-35
enabling filtering … 4-36
filtering … 4-34
manually re-enabling port … 4-36
protection … 4-34
show configured ports … 4-37

Q

QinQ
configuration example … 8-17–8-21
configuring … 8-13–8-23
assigning ports to VLANs … 8-10
auth-vid/unauth-vid … 8-28
changing bridge modes … 8-10, 8-22
creating S-VLANs … 8-14
port S-VLAN membership … 8-15
port trunks … 8-30
ports … 8-13, 8-23
port-types … 8-10, 8-16
qinq modes … 8-7
updating VLAN types … 8-11
VLAN mode … 8-14
VLANs … 8-8
c-tagged interfaces not supported … 8-12
C-VLAN … 8-5, 8-6
definition … 8-6
deleting S-VLANs … 8-14
disabling … 8-7, 8-22
disabling gvrp at interface … 8-15
duplicate VIDs … 8-9
enabling … 8-7, 8-14
erasing configurations when changing modes … 8-22
event log messages … 8-33
features and benefits … 8-5
frame size increases … 8-11
GVRP … 8-15, 8-23, 8-29
disabling at the interface … 8-15
impacts on other features … 8-28–8-32
aaa … 8-28
ACLs … 8-28
arp-protect … 8-28
CDP … 8-29
DHCP … 8-29
directed-broadcast … 8-29
GVRP … 8-29
igmp-proxy … 8-29
IPv6 … 8-29
jumbo frames … 8-29
LACP … 8-23, 8-30
load-sharing … 8-30
meshing … 8-10, 8-12, 8-30
mirroring/monitoring … 8-23, 8-31
multicast-routing … 8-31
QoS … 8-31
routing … 8-31
spanning tree … 8-31

stacking … 8-10, 8-32
UDLD … 8-10, 8-32
Voice VLANs … 8-32
VRRP … 8-32
interoperating with non-ProCurve devices … 8-12
IP support … 8-11
jumbo frames … 8-11
LLDP not supported … 8-30
management VLAN … 8-11, 8-30
meshing … 8-12, 8-30
MIB objects … 8-33
mixed vlan mode … 8-6, 8-7, 8-8, 8-10, 8-14
port
assigning to vlans … 8-15
assignment to VLANs … 8-10
configuration … 8-13
configuring port trunks … 8-30
configuring port-types … 8-16
configuring S-VLAN membership … 8-15, 8-23
configuring uplink ports … 8-19
customer-edge port … 8-12
customer-network port … 8-10, 8-16
customer-network port-type … 8-6
moving from C-VLANs to S-VLANs … 8-23
port-type defaults … 8-10
provider-network port … 8-10, 8-16
provider-network port-type … 8-6
port-types … 8-13
primary VLAN … 8-11
provider core bridge … 8-7, 8-10, 8-17
provider edge bridge … 8-7, 8-10, 8-17
purpose … 8-4
restrictions … 8-10
service provider … 8-4, 8-6
show commands … 8-24
SNMP support … 8-33
software license requirements … 8-3
spanning tree for provider network … 8-20
S-VLAN … 8-5, 8-6, 8-14
S-VLAN bridge … 8-7
svlan mode … 8-7, 8-14
S-VLANs and C-VIDs … 8-18
tag-type, tpid value … 8-12
use model … 8-4
used in a single enterprise … 8-4
VID configuration options … 8-9, 8-11
VLAN configurations … 8-7, 8-8
voice-VLANs not supported … 8-11

Quality of Service

802.1p priority
honored in downstream devices … 6-15
mapped to outbound port queues … 6-15
packet marking … 6-7
basic operation … 6-9
changing the number of outbound queues … 6-96
classifier-based configuration … 6-6, 6-70, 6-71, 9-33
classifier-based override of global configuration … 6-18, 6-81
configuring number of priority queues … 6-95
definition of terms … 6-7
determining outbound port queue … 6-53
DSCP Policy table … 6-89
feature description … 6-4
from upstream device … 6-9
global configuration … 6-19
GVRP not supported … 6-56
in VLAN- and untagged-VLAN environments … 6-17
inbound traffic on network edge … 6-5
marking packets … 6-8, 6-14
match criteria
DSCP codepoint … 6-41, 6-72
IP address … 6-33, 6-72
IP precedence bits … 6-41, 6-72
Layer-3 protocol … 6-54, 6-72
source port … 6-62
TCP/UDP … 6-24, 6-72
VLAN ID … 6-56, 6-72
maximum remarking entries … 6-23
No-override in DSCP Policy table … 6-90
No-override in show qos output … 6-22
on downstream device … 6-7
on edge switches … 6-8
outbound VLAN traffic … 6-5
overview … 6-1
packet classification
classifier-based … 6-7, 6-13
global … 6-12
global configuration … 6-20
overview … 6-12
See also match criteria. … 6-12
packet marking

802.1p priority … 6-14, 6-18
classifier-based QoS … 6-18, 6-71
DSCP codepoint … 6-16, 6-18
global QoS … 6-14, 6-20
IP precedence … 6-18
rate-limit … 6-18
QinQ configuration impacts … 8-31
QoS policy
classifier-based … 6-8, 6-11, 6-73
globally-configured … 6-8, 6-10
queue configuration … 6-95
restrictions
classifier-based QoS … 6-75, 6-85
global QoS … 6-22
show resources command … 6-32, 6-85
viewing configuration … 6-21, 6-81
quick start … 1-7

R

rate-limiting
classifier-based QoS … 6-18, 6-71, 6-77
restrictions … 6-77
redundant links … 5-4, 5-19
non-meshed … 5-18
redundant path … 4-10
region … 4-10
See spanning-tree, 802.1s.
resequencing
class configuration, classifier … 9-23
policy configuration, classifier … 9-29
revision number … 4-14
root-history … 4-62
routing
non-routable VLAN … 2-59
RSTP
meshing requirement … 5-6

S

secure management VLAN … 2-52
setup screen … 1-7
show policy resources command … 6-83, 9-31
show resources command … 6-32, 6-85
show vlan custom … 2-36
single forwarding database … 2-20
single point of failure, removing … 5-2
spanning tree
802.1s
See spanning tree, 802.1s.
blocked link … 4-12
blocked port … 4-10
broadcast storm … 4-3
config-name … 4-47
config-revision … 4-47
enabling MSTP … 4-44
instance vlan … 4-46, 4-47
MSTP
See spanning-tree, 802.1s
root-history … 4-62
VLAN effect on … 2-62
spanning-tree, 802.1s … 4-4, 4-6
802.1D and 802.1w connections … 4-16
802.1D as a region … 4-13, 4-16
802.1Q VLANs … 4-12
802.1s standard-compliant … 4-6
802.1w as a region … 4-13
active path … 4-10
active paths … 4-15
bandwidth loss … 4-12
benefit … 4-6
blocked traffic … 4-11
boundary port, region … 4-14, 4-15
boundary port, VLAN membership … 4-11
BPDU … 4-11, 4-19, 4-23, 4-24, 4-28
BPDU requirement … 4-14
BPDU, function … 4-14
bridge … 4-14
bridge, designated for region … 4-14
caution … 4-6, 4-9
CIST … 4-8, 4-13, 4-15
CIST per-port hello time … 4-15
CIST root … 4-27
CIST root, display change history … 4-62
common and internal spanning tree
See CIST.
common spanning tree
See CST.
compatibility … 4-16
compatibility mode … 4-23
configuration … 4-21, 4-44
configuration identifier … 4-14
configuration steps … 4-19
configuration, BPDU port protection … 4-30
configuration, exchanging … 4-44
configuration, MST instance … 4-39

configuration, MSTI per-port … 4-41
configuration, port … 4-26
CST … 4-8, 4-11, 4-13
CST and legacy devices … 4-11
CST, view status … 4-55, 4-56
debug, display counters … 4-65, 4-66, 4-68
default configuration … 4-9
designated bridge … 4-11, 4-14
designated port … 4-11
disabling MSTP … 4-44
display statistics and configuration … 4-53
dynamic VLANs, disallowed … 4-9
edge port … 4-27
enabling a region … 4-44
enabling MSTP … 4-44
example of multiple topologies … 4-10
fault tolerance … 4-6
force protocol version … 4-16
forward-delay … 4-23
forwarding paths … 4-16
forwarding state … 4-27
frame duplication and misordering … 4-16
general operation … 4-4, 4-6
GVRP … 4-8, 4-15
hello-time, CIST root, propagated … 4-15, 4-24
hello-time, override … 4-15
hello-time, propagated … 4-15
hop-count decremented … 4-24
instance … 4-4, 4-15, 4-20
instance, display debug counters … 4-65, 4-66, 4-68
instance, forwarding topology … 4-16
instance, IST … 4-8
instance, type … 4-8
internal spanning tree
See IST.
interoperating with 802.1D and 802.1w … 4-13
IST … 4-8
IST instance … 4-8, 4-39, 4-48
IST root … 4-8, 4-10, 4-14
IST root, display change history … 4-62
IST, defined … 4-13
IST, dynamic VLAN … 4-15
IST, root switch … 4-13
IST, switch membership … 4-13
IST, VLAN membership … 4-8
legacy devices and the CST … 4-11
legacy STP and RSTP … 4-11
mesh environment … 4-6, 4-16
MIB … 4-53
MST region
See region.
MSTI … 4-8, 4-15
MSTI root … 4-10
MSTI root, display change history … 4-62
MSTI, view status … 4-57
MSTP … 4-9
MSTP operation … 4-9
MSTP, view global configuration … 4-58
multiple spanning tree instance
See MSTI
override hello-time … 4-15
path cost, effect on 802.1D … 4-16
pending configuration … 4-61
pending option … 4-9, 4-22, 4-44
per-VLAN STP … 4-6
planning … 4-17
port connectivity … 4-26
port states … 4-10, 4-16
priority resolution … 4-40
priority, device … 4-19, 4-25
priority, IST port … 4-43
priority, MSTI port … 4-42
QinQ configuration impacts … 8-31
rapid state transitions … 4-16
redundant links … 4-11
region … 4-4, 4-7, 4-8, 4-9
region name … 4-14, 4-21
region root switch … 4-8
region, configuration name … 4-73
region, Configuration Revision number … 4-73
region, defined … 4-14
region, enabling … 4-44
region, root bridge … 4-13
region, RSTP bridge … 4-15
region, switch configuration … 4-15
region, switch excluded … 4-73
region, view configuration … 4-60
region, VLAN assignments … 4-14
regional boundary port … 4-14
regional root bridge per-instance … 4-11
regional root switch … 4-13
regional root switch, configuration … 4-15
regions, communication between … 4-15
root bridge … 4-8
root bridge per-instance … 4-11

root bridge per-region … 4-13
root port per-instance … 4-11
root switch, instance … 4-40
root switch, IST instance … 4-8, 4-13
root switch, MST instance … 4-15
root switch, regional … 4-13
root, CIST … 4-24
root, IST … 4-14
root, MSTI … 4-10
routed traffic in a region … 4-11
RSTP as a region … 4-7
RSTP BPDU requirement … 4-14
RSTP bridge … 4-15
rules for operation … 4-15
separate forwarding paths … 4-8
show commands … 4-53, 4-62
SNMP MIB … 4-53
STP as a region … 4-7
switch excluded from region … 4-73
topology between regions … 4-10
troubleshooting … 4-62
trunk, root, per-instance … 4-11
trunked link … 4-58
trunked link example … 4-12
types of MST instances … 4-8
VLAN assignments, region … 4-14, 4-15
VLAN membership, region … 4-12
VLAN, change instance … 4-20
VLAN, configuration error … 4-73
VLAN, connectivity between regions … 4-15
VLAN, duplicate or missing packets … 4-73
VLAN, dynamic … 4-8
VLAN, instance assigned … 4-10, 4-15, 4-39, 4-48
with legacy STP and RSTP … 4-7

stacking
benefits … 7-3
minimum software version, other ProCurve switches … 7-9
primary … 7-45
QinQ support … 8-32
See also virtual stacking.

static VLAN, convert to … 3-4

STP
cost change by mesh switch … 5-19

subnet address … 2-8

switch meshing
See mesh.

T

TCP/UDP
match criteria, classifier … 6-72, 9-17
operators … 9-17, 9-18
packet classification, global QoS … 6-24
well-known port names … 9-18

ToS
compared to IPv6 Traffic Class byte … 9-11
See Type of Service.

Traffic Class byte
compared to IPv4 ToS byte … 6-43
defined … 6-9
in IPv6 header … 9-11

trunk
spanning-tree example … 4-12

Type of Service
defined … 6-9
determining outbound port queue … 6-53
DTR bits as match criteria, classifier … 9-10
in IPv4 header … 9-11
IPv4 ToS compared to IPv6 Traffic Class byte … 9-11
match criteria, global QoS … 6-41
ToS byte compared to IPv6 Traffic Class byte … 6-43

Type, meshed port … 5-10

U

unicast
in switch mesh … 5-17

upstream device (QoS)
definition … 6-9

V

VID
See VLAN.

virtual stacking
transmission interval range … 7-16

VLAN
802.1Q VLAN in mesh … 5-20
already exists, message … 2-44
broadcast domain … 2-5
CLI, commands … 2-31
CLI, configuring parameters … 2-30
convert dynamic to static … 2-43, 3-4
custom display fields … 2-37

customer VLAN
See C-VLAN
customizing output … 2-36
C-VLAN (QinQ) … 8-5, 8-6
dedicated management … 2-51
default VLAN VID … 2-51
default VLAN, name change … 2-51
DEFAULT_ VLAN … 2-51
deleting … 2-16, 2-41, 2-64
deleting, with member ports … 2-16, 2-41, 2-42
DHCP, primary VLAN … 2-51
double-tagging … 8-11
duplicate MAC address … 2-20
dynamic … 2-5, 2-19, 2-24, 2-30, 2-43, 4-15
effect on spanning tree … 2-62
gateway, IP … 2-52
GVRP, auto … 2-15
heartbeat packets, configuring … 2-67
IP interface relationship … 2-63
layer-2 broadcast domain … 2-6
layer-3 broadcast domain … 2-6
limit … 2-24, 2-30
MAC address assignment … 2-63
MAC address reconfiguration … 2-65
MAC address, verifying … 2-69
match criteria, classifier … 6-72, 9-11
match criteria, global QoS … 6-56
maximum, GVRP … 3-18
menu, configuring parameters … 2-24
menu, maximum capacity … 2-28
menu, missing VLAN … 2-28
mesh domain and … 5-20
migrating layer-3 VLANs … 2-65
mixed vlan mode (QinQ) … 8-6
multiple forwarding database … 2-20, 2-23
multiple in switch mesh … 5-17
multiple VLANs on port … 2-48
non-routable … 2-59
number allowed, including dynamic … 2-28
per port configuration options … 2-14
port assignment … 2-28
port configuration … 2-50
port monitoring … 2-63
port restriction … 2-64
port trunk … 2-63
port-based … 2-6
primary … 2-40, 2-51, 7-9, 7-33, 7-45
primary, CLI command … 2-31, 2-39
primary, select in menu … 2-25
primary, web configure … 2-45
primary, with DHCP … 2-15
protocol … 2-6, 2-7, 2-11, 2-15, 2-17, 2-63
ARP requirement … 2-15, 2-41
capacity per VLAN … 2-15
CLI only … 2-24
commands … 2-31
compared to port-based … 2-8
configuration … 2-30, 2-41
example … 2-49
forbid option not allowed … 2-44
IP addressing … 2-8
IPv4 routing … 2-9
IPv4, ARP requirement … 2-15, 2-41
IPv6 … 2-8
limit … 2-14
limit on types per-port … 2-9
non-routable … 2-9, 2-12, 2-46
operation … 2-17
port membership limit … 2-9
primary VLAN not allowed … 2-40, 2-52
router, external … 2-10, 2-12, 2-64
routing … 2-6, 2-10, 2-64
status … 2-32, 2-33, 2-35
tagged … 2-14, 2-48
tagged member … 2-9
tagging … 2-10
traffic separation … 2-5
types … 2-11, 2-41
untagged member … 2-9
untagged packet forwarding … 2-16
untagged, limit … 2-14
untagged, multiple … 2-48
untagged, restriction … 2-64
QinQ configuration options … 8-7, 8-8
restrictions … 2-64
routing between VLANs … 2-5
routing, protocol VLANs … 2-6
secure management … 2-52
security, network … 2-5
service VLAN
See S-VLAN
show vlan ports detail … 2-32
single forwarding database … 2-20
static … 2-5, 2-7, 2-24, 2-30, 2-52
static, in switch mesh … 5-6
subnet … 2-5

S-VLAN (QinQ) … 8-5, 8-6
svlan mode (QinQ) … 8-7
switch capacity … 2-5
switch mesh … 5-6
tagging … 2-46, 2-48
tunnel VLAN
See S-VLAN
unknown VLAN … 3-11
untagged … 2-13, 2-29
untagged, operation … 2-17
VID … 2-5, 2-48
VID, default VLAN … 2-52
voice … 2-6, 2-32, 2-33, 2-35, 2-62
voice, configuration … 2-42
voice, configuring … 2-31
voice, VLAN type … 2-15
web browser configuration … 2-45
See also GVRP.
VLANs
static, 802.1s spanning tree … 4-8
voice VLAN
See VLAN.
VoIP
operating rules … 2-15
See also VLAN, voice.
VPN solution for MANs
QinQ … 8-5

W

warranty … -ii
web browser interface
configuring VLANs … 2-45
write memory
converting dynamic to static VLAN … 3-18

Technology for better business outcomes

To learn more, visit www.hp.com/go/procurve/

© Copyright 2010 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP will not be liable for technical or editorial errors or omissions contained herein.

5992-3060, March 2010